Chapter 22

Quantum Complexity

May 6, 2002

Quantum computation is an active topic of current research. It is fundamentally different from classical computation because different laws of physics are used. The idea of quantum computing was first suggested in the early 1980's by Paul Benioff [3, 4, 5], a physicist from Argonne National Laboratory. Feynman [17] showed how to build a classical computer based on quantum principles. Deutsch [14, 15] introduced the universal quantum computer and quantum circuits. An comprehensive treatment of quantum computation including quantum information theory is Nielsen and Chuang [21].

Related questions about the physics of information and computing date back earlier: since information is physically represented, what does the laws of physics say about fundamental limits of computation? Rolf Landauer (1927-1999) was interested in minimum energy computation. He noted (1961) that erasure of information is necessarily a dissipative process – heat is lost from the computer into the environment. Thus, if we erase one bit of information, the overall entropy increases by $k \ln 2$. At temperature T, the work expended is $kT \ln 2$. The converse to this so-called "Landauer Principle" is that if we compute reversibly (with no erasure of information) then no dissipation or power consumption is needed. In 1973, Bennett [7] showed that such computation is always possible in theory (but it would be a very slow computation!). Reversible computation was further investigated by Tofolli and Fredkin [24, 18]. Although reversible computing is based on classical laws of physics, it can be viewed as a precursor to quantum computation. For a history of reversible computing, see [8].

Two developments in the early 1990s help to push quantum computing out of the curiosity stage. One was the discovery of quantum algorithms that have major implications for cryptography. In 1994, Peter Shor at AT&T Labs showed that the problems of integer factorization and discrete logarithm can be solved by quantum computers with high probability in polynomial time [23]. As it is widely believed that both these problems are non-polynomial time on classical computers, and the security of many cryptographic protocols depend on these assumptions, this suggests a "killer app" for quantum computing. As a result, the subject holds real interest for agencies such as DARPA and the National Security Agency. The other development is the experimental demonstration of techniques that could be used to build quantum computers. A major challenge here is to isolate the quantum bits (qubits) from environment (to keep the system "coherent"). Several competing technologies are being investigated. Seth Lloyd (1993) showed that a quantum computer could be built from an array of coupled two-state quantum systems. An implementation proposed by Chuan and Gershenfeld, and independently by Cory, Fahmy and Havel, is based on spins in the nucleus of atoms. Such nuclear qubits are naturally isolated from the external world by its clouds of electrons, and they may be assembled naturally as molecues. The technology for manipulating nuclear spins is NMR (nuclear magnetic resonance), a well-developed technology routinely used in medicine. In August of 2000, a 5-qubit computer was announced by IBM corporation. The ion trap approach of Ignacio Cirac and Peter Zoller [12] is based on confining cold ions along a line ("linear Paul trap"). The quantum state of each ion is a superposition of its ground state $|0\rangle$ and some relatively long-lived excited state $|1\rangle$, representing the qubits (see below). Laser beams directed at the individual ions can achieve the transitions within each ion but how can the qubits interact in the quantum mechanical sense? Cirac and Zoller showed with proper tuning of the lasers, the controlled XOR gate (see below) can be implemented with 5 laser pulses. Such devices have been constructed [22, 20]. The speed of such a device depends on the frequency of the fundamental vibrational modes of the ions; current technology can perhaps achieve 10^4 steps/second (see [2]). Even if current approaches do not lead realistic quantum computers, they are nevertheless useful for demonstrating the principles of quantum computation [2].

Quantum computing also has implications for the fields of information theory, coding theory and cryptography. See the survey [10]. It calls for new foundations for each of these areas. Among other things, one goal in this chapter is to present Shor's algorithm for factoring.

22.1 Quantum Computing

22.1.1 Quantum Bits

The basic unit of information in a classical computers is the **bit**, an entity that can assume exactly one of two distinct values, denoted 0 and 1. The analogous **quantum bit** (or "qubit") also has two special values (the eigenvectors or pure quantum states) which we identify with the classical values 0 and 1. Following a standard notation[1] in physics these **pure states** or **eigenstates** are denoted $|0\rangle$ and $|1\rangle$, respectivly. In general, if Ψ is the "name" of a quantum state, we write $|\Phi\rangle$ to denote the state. What we choose for the name is not important. For instance, physicists often use suggestive symbols such as $|\uparrow\rangle, |\downarrow\rangle$ for these states. However, the value of a qubit is a **quantum state** of the form

$$c_0|0\rangle + c_1|1\rangle \tag{1}$$

where $c_0, c_1 \in \mathbb{C}$ (complex numbers) satisfying $|c_0|^2 + |c_1|^2 = 1$. For instance, if $c_0 = 0, c_1 = 1$, then the quantum state is just $|1\rangle$, a pure state. We say that the quantum state is a **superposition** of the pure states.

Mathematically, a pure state is just a basis vector in some chosen basis. Thus the states of a qubit lives in the complex two-dimensional vector space in which all the vectors have unit length. We could discuss all our results using only this mathematical model, but in the following we will suggest some physical intuitions and possible interpretations of the mathematics.

The qubit can be realized by a variety of physical quantum systems. Any of these systems, in principle, can be the basis for constructing quantum computers. For instance, an electron or other spin-$\frac{1}{2}$ particle can have one of two distinct spins, called spin-up and spin-down. Thus the state of a qubit can be encoded by the state of an electron spin. Alternatively, the photon (light particle) is a massless, spin-1 particle that can have one of two independent polarizations. Using a photon as a qubit, we can manipulate its state by rotating the polarization of the photon.

How does a qubit relate to a classical bit? **Measurement** is an operation applicable to quantum states. When the quantum state in (1) is measured, the quantum state "collapses" to $|i\rangle$ with probability $|c_i|^2$ where $i = 0, 1$. Thus, unlike classical bits, which can be measured without affecting its value, any measurement of qubits is potentially distructive. The classical analogue of the quantum state (1) is the **random bit** which assumes the value of 0 with probability $|c_0|^2$ and assumes the value of 1 with probability $|c_1|^2$. But they are not equivalent: using classical random bits, we cannot distinguish between $c_0|0\rangle + c_1|1\rangle$ and $c_0|0\rangle - c_1|1\rangle$. We describe some physical experiments to clarify this.

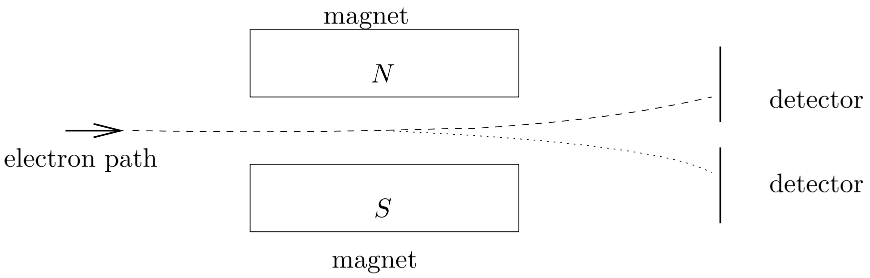


Figure 22.1: Stern-Gerlach measuring device

The physical interpretation of measurement in spin models is illustrated in an apparatus known as a Stern-Gerlach device. See Figure 22.1 for a 2-dimensional rendition. The apparatus comprises a pair of magnets that surrounds the path of an electron, with two detector plates at the exit paths of the electron. When an electron with an up- or down-spin passes through the apparatus, the path of the electron will deflect up or down, depending on the spin. In general, an electron is in a supposition of up- and down-spins and it will deflect in either direction, with a suitable probability. This device illustrates the idea that measurements are relative to a choice of basis: the magnetic flux is conventionally said to flow in the horizontal direction, so that the electron deflects up or down. In Figure 22.1, this up/down direction corresponds to the vertical paper direction But if we rotate the device about the axis of the path, the results of measurement would be different: if we rotate it by 90°, the same electron would now deflect left or right with suitable probability. In Figure 22.1, left/right means into/out of the page. This amounts to a new measurement basis. We normally have the freedom to choose a basis that is most convenient for the application.

[1]This is the **ket** notation; there is a corresponding **bra** notation which has the form $\langle y|$. The pair $|x\rangle$ and $\langle y|$ can be composed to $\langle y|x\rangle$ which can be viewed as the scalar product of two vectors. The "bra-ket notation" is from the physicist Dirac.

Suppose we pass the up-spin electrons through a second Stern-Gerlach device where magnets are now rotated by 90° to cause a left-right deflection. We will see that again, the left- and right-spin electrons are equally probable. This fact might be surprising if we expect the up-spin electrons not to have any left- or right-spin components. But perhaps electrons have independent spin components for up/down as well as left/right spins. But if we continue by passing the left-spin electrons through a third Stern-Gerlach device with the up-down orientation as the first device, we again see an equal probability of the electrons deflecting up or down. This is a surprise if we had expected to see up-spins only, as only up-spin electrons were sent through the second device. To explain this, we postulate that the pure states in the basis of the up/down measurement device are $|up\rangle$ and $|down\rangle$, respectively. But relative to the left/right measurement, the pure states are $|left\rangle = (|up\rangle + |down\rangle)/\sqrt{2}$ and $|right\rangle = (|up\rangle - |down\rangle)/\sqrt{2}$. Thus the electrons sent into the second device are in state $|up\rangle$, but in the measurement basis, this appears as $(|left\rangle + |right\rangle)/\sqrt{2}$, and thus they have equal probability of going left or right. The electrons sent into the third device are in the state $|left\rangle$, but relative to the measurement basis of left/right, they have equal probabilities of going left or right.

Superposition is different from the probabilistic phenomenon of being in one of several states with some probability. A series of experiments[2] based on the Mach-Zehnder interferometer illustrate this. Refer to Figure 22.2. In experiment A, we reflect a photon off a half-silvered mirror M_1, we will detect the photon at detector 1 or 2 with 50% probability each. The classical explanation is that the photon has equal probability of taking either path. The quantum mechanical explanation is that both paths are taken at once (superposition), but the detectors cause a collapse of the state. But so far, we have no basis to prefer one explanation over the other (in fact, the classical one should be preferred for its simplicity). But in experiment B, we place two fully silvered mirror at

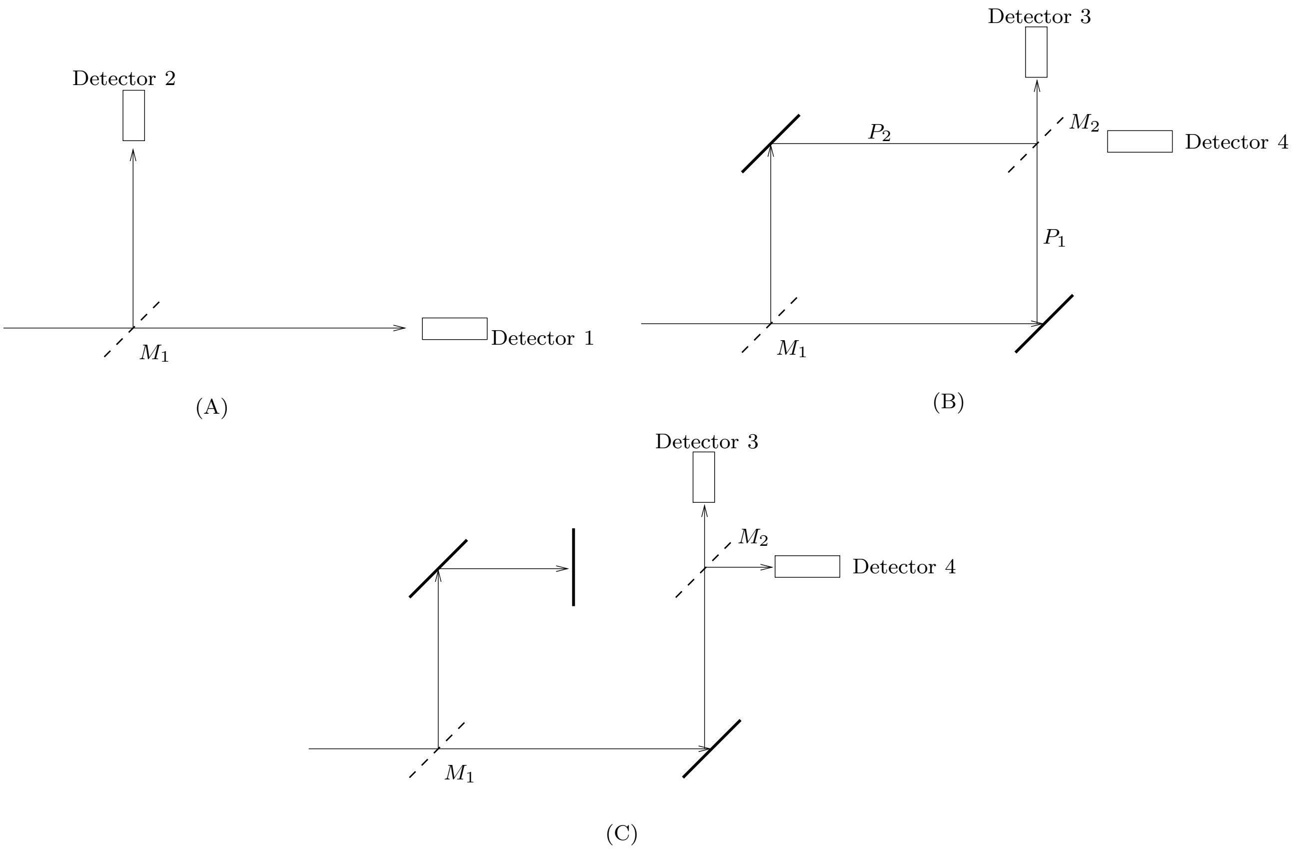


Figure 22.2: Experiments A, B, C.

the positions of detectors 1 and 2, but arranged so that the two photon paths (P_1, P_2) recombine. We place a second half-silvered mirror M_2 at the point of recombination. We also place the detectors 3 and 4 to measure the reflection or non-reflection from M_2. It turns out, the photon reaches detector 3 with 100% probability and never reach detector 4. This is impossible to explain classically. The quantum mechanical view accounts for this: the photon must have travelled both paths P_1 and P_2, and when recombined, it is able to distinguish the two choices

[2]See "Un saut d'echelle pour les calculateurs", by A. Barenco, A.Ekert, A. Sanpera and C.Machiavello, in La Recherche, Nov 1996. Adapted article by Barenco may be found in http://www.qubit.org/intros/comp/comp.html. See also [13].

at M_2 and only chose (by way of interference) the "correct path". In experiment C, we confirm this explanation by placing a barrier B in path P_2 of the previous experiment. Now, we have 50% probability of detecting the photon at detectors 3 and 4.

22.1.2 Quantum Words

Real world computers operate on fixed size sequence of bits, called a **word**. A word in modern computers is typically 32 bits or 64 bits long. Similarly, a finite sequence of qubits will be called a **quantum word** ("quword" for short). In the literature, a quword is also known as a "quantum register". An array of n qubits is called a n-quword.

Classically, transition from bits to words is trivial. But quantum words introduce a new situation with no classical analogue. This is the phenomenon of quantum interference and phase information. This is already evident for $n = 2$. Let A and B be the qubits in a 2-quword. If A is in the pure state $|0\rangle$ and B in $|1\rangle$, then the state of the quword is written $|0\rangle \otimes |1\rangle$ (tensor product), or more compactly, $|01\rangle$ and sometimes $|0\rangle|1\rangle$. This looks like Cartesian product, but tensor product is more than this. And hint that something else is going on is the following basic property of tensors which we will use often: for any scalar α,

$$\alpha(u_1 \otimes u_2) = (\alpha u_1) \otimes u_2 = u_1 \otimes (\alpha u_2). \tag{2}$$

If A, B are in the superposition of pure states $x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $y = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, respectively, then their joint state is given by

$$|x\rangle \otimes |y\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle).$$

Next supposed that the quword is in the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ ("Bell State"). These two qubits are "entangled": when you measure one of the qubits, then the other qubit would also collapse to the same value. Intuitively, they are maximally entangled (or correlated); quantum information theory is the subject that shed light on this phenomenon.

In general, an n-quword has 2^n pure states of the form $|b_1 \cdots b_n\rangle$ where $b_i \in \{0, 1\}$. For instance, if $n = 2$, then the possible pure states are $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Interpreting $b_1 \cdots b_n$ as a binary representation of a natural number, the pure states can be denoted $\{|i\rangle : i = 0, 1, \ldots, 2^n - 1\}$. A quantum state is again a superposition of these pure states, and can be represented by a unit length vector in $c \in \mathbb{C}^{2^n}$. Unit length means that $\sum_{i=0}^{2^n-1} |c_i|^2 = 1$ where $c = (c_0, \ldots, c_{2^n-1})$. The quantum state corresponding to c is $\sum_{i=0}^{2^n-1} c_i|i\rangle$. Note that we have just established 2 conventions for writing pure states: $|x\rangle$ where x is either a binary string or a natural number. In practice, these mutually exclusive conventions are use interchangeably when there is no confusion. For bit strings x, y representing two disjoint quwords, the ket-notation admits the operation $|x\rangle \otimes |y\rangle = |xy\rangle$. When x, y are superposition of states for two disjoint quwords, $|x\rangle \otimes |y\rangle$ is obtained as a Cartesian product of the separate pure states.

Quwords offer a twist to the concept of measurement. Naturally, if $|x\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle$ and it is measured, then the state collapses to $|i\rangle$ with probability $|c_i|^2$ for each i. But we can also measure any individual qubit, or more generally, any subset Y of qubits. If the set of n qubits is X, let $Z = X \setminus Y$ be the complementary set of qubits. Each pure state $|i\rangle$ of X can be written as $|y\rangle \otimes |z\rangle$ where $|y\rangle, |z\rangle$ are pure states of Y and Z. Denote the Y-projection of $|i\rangle$ by $\pi_Y(|i\rangle)$, and thus

$$|i\rangle = \pi_Y(|i\rangle) \otimes \pi_Z(|i\rangle).$$

Let $P_Y(y) = \{i = 0, \ldots, 2^n - 1 : \pi_Y(|i\rangle) = |y\rangle\}$. When we measure the Y-bits of a state $|x\rangle$, we will see each pure state $|y\rangle$ of Y with probability $p(y) := \sum_{i \in P_Y(y)} |c_i|^2$. After a measurement which revealed a particular $|y\rangle$ in Y, the state of the quword is given by

$$\frac{\sum_{i \in P_Y(y)} : c_i|i\rangle}{\sum_{i \in P_Y(y)} |c_i|^2}.$$

Let us illustrate this for a 3-quword whose pure states are labeled $|0\rangle, \ldots, |7\rangle$ as usual. If Y refer to the middle bit, then $P_Y(0) = \{0, 1, 4, 5\}$ and $P_Y(1) = \{2, 3, 6, 7\}$. Upon measuring the y-bit of state $|x\rangle = \sum_{i=0}^{7} c_i|i\rangle$, we see the value of $|0\rangle$ with probability $p(0) = |\alpha_0|^2 + |\alpha_1|^2 + |\alpha_4|^2 + |\alpha_5|^2$. The new state of the quword is $(\alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_4|4\rangle + \alpha_5|5\rangle)/p(0)$. There is an analogous case where we see the value $|1\rangle$ for the Y-bit.

The general treatment of measurements is in terms of **measurement operator**, $M = \{M_i : i \in I\}$ where each M_i is an "outcome operator", and I is the set of possible outcomes of the measurement M. The outcome operators satisfy the equation

$$\sum_{i \in I} M_i^* M_i = I. \tag{3}$$

The probability of outcome $i \in I$ when M measures a state $|x\rangle$ is given by

$$p(i) := \langle x|M_i^* M_i|x\rangle$$

and the corresponding collapsed state is

$$\frac{M_i|x\rangle}{\langle x|M_i^* M_i|x\rangle}. = \frac{M_i|x\rangle}{p(i)}.$$

The Equation (3) simply ensures that

$$\sum_{i\in I} p(i) = \sum_{i\in I} \langle x|M_i^* M_i|x\rangle = 1.$$

22.1.3 Axiomatic Quantum Mechanics

The simplest approach to quantum mechanics is an axiomatic one. We postulate the basic entities, operators and their properties, and investigate them mathematically. The physical interpretations follows, hopefully, with some guidance. As seen above, quantum states are some kind of complex vectors. Following von Neumann, we can postulate the space of quantum states as a suitable Hilbert space. See the appendix for additional mathematical background.

Let S be a complex vector space is endowed with an inner product $\langle\cdot,\cdot\rangle : S \times S \to \mathbb{C}$. The space S can be finite dimensional or infinite dimensional. Elements $\varphi, \psi \in S$ of S are called **states**. The Dirac notation favored by physicists will write "$|\varphi\rangle$" instead of a plain φ. Each φ defines a dual vector (see below) which in the Dirac notation is written "$\langle varphi|$" (the bra-notation). Consistent with this notation, the inner product $\langle\psi, \varphi\rangle$ may be written "$\langle\psi|\varphi\rangle$", viewed as a product of the Dirac bra- and a ket-notation. We emphasize that these are stylistic conventions.

Define the **norm** of $\varphi \in S$ by $\|\varphi\| := \sqrt{\langle\varphi|\varphi\rangle}$. A sequence $\{x_0, x_1, \ldots\}$ in S with respect to the norm $\|\cdot\|$ is **Cauchy** if for every $\varepsilon > 0$ there exists n such that $\|x_i - x_j\| < \varepsilon$ whenever $i, j > n$. We say S is **complete** relative to this norm if every Cauchy sequence $\{x_0, x_1, \ldots\}$ in S with respect to the norm converges to an element of S. The inner product satisfies the axioms:

1. (Positivity) $\langle\psi|\psi\rangle$ is real and non-negative for all ψ. Furthermore, it is strictly positive iff $\psi \neq 0$.
2. (Linearity) $\langle\psi|a\varphi_1 + b\varphi_2\rangle = a\langle\psi|\varphi_1\rangle + b\langle\psi|\varphi_2\rangle$.
3. (Skew Symmetry) $\langle\psi|\varphi\rangle = \langle\varphi|\psi\rangle^*$.

If S satisfies the above properties, it is called a **Hilbert space**. Hilbert spaces has a natural topology induced by the metric $d(x, y) = \|x - y\|$. So the notion of continuity of a function $f : S \to \mathbb{C}$, etc, is meaningful.

Examples: Let $S = \ell_2(n)$ denote the vector space $\mathbb{C}^n$ where $x, y \in \mathbb{C}^n$ has inner product $\langle x|y\rangle = \sum_{i=1}^n x_i^* y_i$, x_i, y_i being the components of x, y, respectively. We can extend this to the infinite dimensional space $S = \ell_2(\infty)$ comprising vectors $(x_i)_{i=0}^\infty$ with $\sum_{i=1}^\infty |x_i|^2 < \infty$. Another infinite dimensional space is $L_2(a, b)$ for reals a, b with $-\infty \leq a < b \leq \infty$. This space comprises all $f : (a, b) \to \mathbb{C}$ such that $\int_a^b |f(t)|^2 dt$ is defined and $< \infty$. The elements in $L_2(a, b)$ are equivalence classes of functions that are almost-everywhere equal. Also $\langle f|g\rangle = \int_a^b f^*(t)g(t)dt$.

For our purposes, it is sufficient to focus on the finite dimensional case. Then S may be identified with subsets of $\mathbb{C}^n$ relative to some basis set $e_1, \ldots, e_m$ for S. States of S have the form $\psi = \sum_{i=1}^m c_i e_i$ where $\sum_{i=1}^m |c_i|^2 = 1$. We postulate the ability to "measure" states. When ψ is measured, it collapses to each e_i with probability $|c_i|^2$. To make the connection to classical computing, the e_i's are classical states.

Linear Functionals. A **linear functional** over S is a continuous linear function $L : S \to \mathbb{C}$. Linearity means $L(ax + by) = aL(x) + bL(y)$ for all $x, y \in S$ and $a, b \in \mathbb{C}$. For any $x \in S$, we can obtain a linear functional $L_x : S \mapsto \mathbb{C}$ where $L_x(y) = \langle x|y\rangle$. A basic result about Hilbert space is that *all* linear functionals are of this form. Let S^* be the space of all linear functionals over S. We make S^* a vector space by defining $aL_x + bL_y = L_{ax+by}$ for all $x, y \in S$ and $a, b \in \mathbb{C}$. This S^* is also called the **dual space** of S. If a state y is written $|y\rangle$ and the linear functional defined by x denoted $\langle x|$, then applying L_x to y yields $\langle x|y\rangle$.

Linear Operators. A **linear operator** of S is a linear function $A : S \to S$. Linearity means $A(ax + by) = A(ax) + A(by)$. When S is finite dimensional, A can be represented by an $n \times n$ matrix. Hence the study of linear operators in these cases reduces to the study of matrix transformations. See the appendix for more information.

Tensor Products. Given two finite dimensional Hilbert spaces S, T, we want to define a new Hilbert space $S \otimes T$ called their **tensor product**. We could define this axiomatically, but it is easiest to define this in the concrete setting of a n-vectors: if $x \in \mathbb{C}^m$ and $y \in \mathbb{C}^n$, then their tensor product is $x \otimes y = (x_i y_j : i = 1, \ldots, m, j = 1, \ldots, n) \in \mathbb{C}^{mn}$. Note that when we write out $x \otimes y$ as a mn-vector, $z = (z_1, \ldots, z_{mn})$, we must have some fixed convention for identifying each component z_k with some $x_i y_j$. That is, we need a bijection $b : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \{1, \ldots, mn\}$ so that $z_k = x_i y_j$ if $b(i, j) = k$. A standard bijection is $b(i, j) = (i - 1)n + j$.

Thus, the property (2) above is easily verified. Also, we have

$$(x + y) \otimes z = xz \otimes yz, \qquad z \otimes (x + y) = zx \otimes zy$$

Associativity of tensor products is clear. In terms of the state notation, we tend to write $|xy\rangle$ instead of $|x\rangle \otimes |y\rangle$. If H_i $(i = 1, \ldots, k)$ are Hilbert spaces, so is $H = H_1 \otimes H_2 \otimes \cdots \otimes H_k$. Moreover, if B_i is an orthonormal basis for H_i, then $B_1 \otimes B_2 \otimes \cdots \otimes B_k$ is an orthonormal basis for H.

In the concrete setting of n-vectors, a linear transformation of S is represented by an $m \times m$ matrix A. If B is a $n \times n$ matrix that represents a linear transformation of T, then we can define their **tensor product** $A \otimes B$ which is a $mn \times mn$ matrix that is a linear transformation of $S \otimes T$. With suitable conventions, $A \otimes B$ can be written

22.1.4 Reversible Circuits

The next topic we address is how quantum states are transformed. The short answer is "by unitary operators on Hilbert spaces". But before we go into such operators, we first treat the intermediate but important concept of reversibility.

Classical states are transformed by Boolean gates. The analoguous **quantum gate** must act "unitarily". Recall[3] that an $m \times m$ matrix U with complex number entries is **unitary** if its inverse is given by its conjugate transpose $U^{-1} = U^*$. A unitary transformation for n-qubit word is represented by a $2^n \times 2^n$ unitary matrix. Since unitary transformations are reversible, this is one motivation to first consider the intermediate concept of reversible transformations. As noted in the introduction, the study of reversible computation actually predates quantum computing. Another rational for introducing reversible logic in quantum computing is based on the fact that *any reversible computation can be simulation on a quantum computer with essentially the same complexity.* So an upper bound on the reversible complexity of a problem yields an upper bound of its quantum complexity. This is useful because reversible computation, being classical, is easier to understand.

A circuit in classical Boolean circuit theory is a directed acyclic graph whose source nodes are called **input nodes** and the rest are called **gates**. Gates are Boolean functions taken from some finite set (the basis). A basis B is **universal** if all Boolean functions can be computed by a circuit based on B. The edges connecting nodes are called "wires" or **lines**; each line carries a bit signal. Usually, there is no restriction on the fanout of inputs or gates. In reversible circuits, extra fanouts are not allowed. A well-known universal basis is the set $\{AND, OR, NOT\}$ (in symbols, $\{\wedge, \vee, \neg\}$). Another universal basis is the singleton $\{NAND\}$ where $NAND(x, y) = \neg(x \wedge y)$.

A reversible function $f : \mathbb{B}^n \to \mathbb{B}^m$ is simply a $1 - 1$ function. Thus $n \leq m$. Often, we might as well assume $m = n$, in which case f is a permutation of $\mathbb{B}^n$. If f is not reversible, we can convert it to a related reversible function f'. The idea is simply to reproduce the input in the output. The function f' is

$$f' : \mathbb{B}^n \to \mathbb{B}^{m+n}$$

where $f'(x) = (x, f(x))$. Clearly, f is reducible to f' provided we assume that certain output lines can be ignored (the n output lines containing x). If we want a permutation, we can also define

$$f'' : \mathbb{B}^{m+n} \to \mathbb{B}^{m+n}$$

where $f''(x, 0^m) = (x, f(x))$. Again, we can reduce f to f'' provided that we can fix certain input lines (setting the last m lines to 0) and ignore certain output lines, as before. These 2 techniques (fixing input lines and ignoring output lines) will be fundamental in the construction of quantum circuits.

The classical XOR gate is defined as follows: $XOR(x, y) = 0$ iff $x = y$. In infix notation, we write $x \oplus y$ for $XOR(x, y)$. It may be easier to understand XOR as addition modulo 2 (the symbol $\oplus$ is highly suggestive of this). This gate is clearly non-reversible. A simple variation gives us the **reversible exclusive-or** (denoted $T_2(x, y)$) with 2-inputs and 2-outputs where

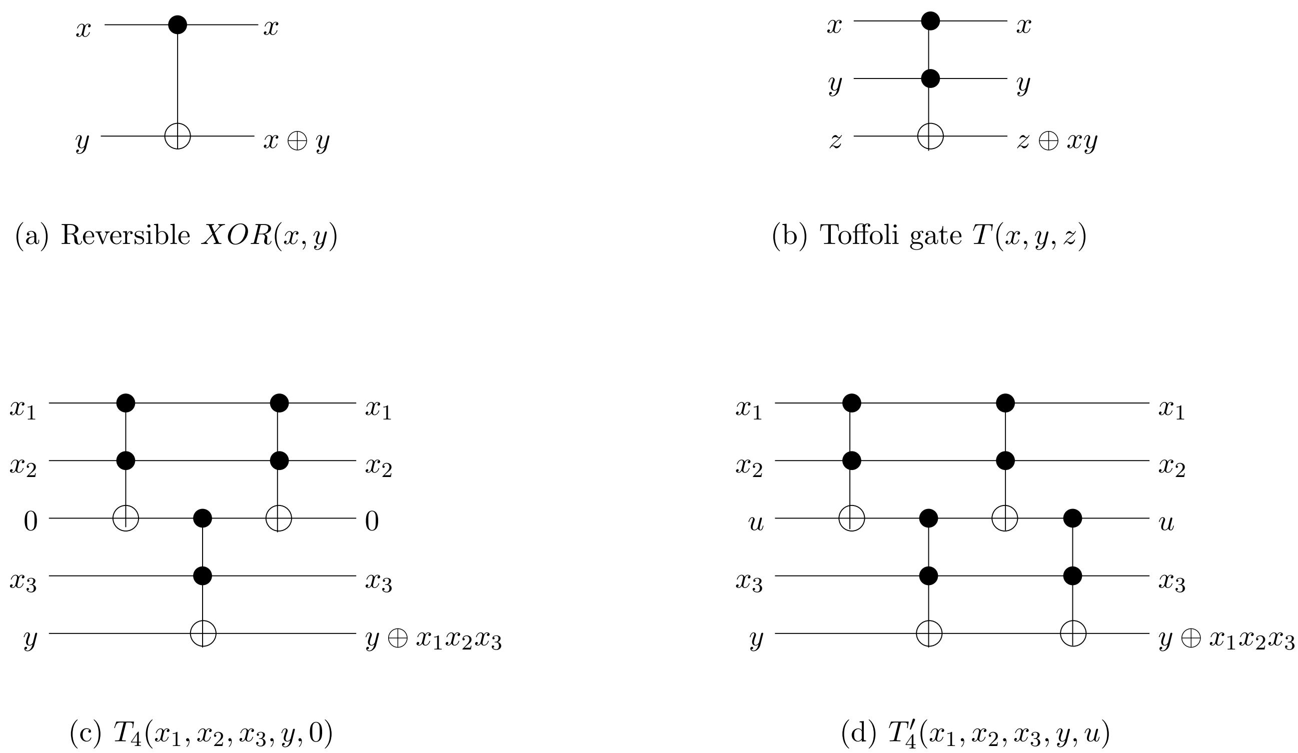
$$T_2(x, y) = (x, x \oplus y).$$

This is also called the controlled-NOT gate as we can think of the first bit as a "control bit", and the second bit as "data bit", which is negated or not, depending on the control bit. The diagramatic representation[4] of this gate

[3] See the appendix.

[4] Feynman [17] introduced this notation, but with the $\oplus$ node is written as an "X".

is seen in Figure 22.3(a). There is a standard convention for reversible (and quantum) circuits seen in figure 22.3: the lines are drawn horizontally and the implicit flow of data in the lines is from left to right.



(a) Reversible $XOR(x, y)$

(b) Toffoli gate $T(x, y, z)$

(c) $T_4(x_1, x_2, x_3, y, 0)$

(d) $T_4'(x_1, x_2, x_3, y, u)$

Figure 22.3: (a) Controlled XOR, $T_2(x, y)$, (b) Toffoli Gate, $T(x, y, z)$, (c) Circuit for T_4, (d) Alternative circuit for T_4

One use of the T_2-gate is as a "copier". If the gate y is prepared as 0, then the $x \oplus y$-output line will contain the input value of x:

$$(x, 0) \mapsto T_2(x, 0) = (x, x). \tag{4}$$

Another application of the T_2-gate is for exchanging any two bits. This is done by arranging three of these gates

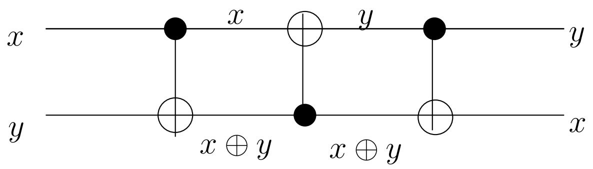


Figure 22.4: Exchange of x and y-bits.

in series, as in Figure 22.4 with the middle gate exchanging the roles of the control bit and data bit:

$$(x, y) \mapsto (x, x \oplus y) \mapsto (y, x \oplus y) \mapsto (y, x). \tag{5}$$

It follows that any permutation of the input bits can be achieved using this gate, as any permutation can be obtained as a product of transpositions (pairwise exchanges). Note that being able to permute bits does not mean the T_2-gate is universal: there are $n!$ bit functions that permute bits of $\mathbb{B}^n$ but there are $(2^n)!$ permutations of the set $\mathbb{B}^n$. Indeed, it can be shown that the set of all 1-bit and 2-bit reversible gates is not universal (see Exercise).

We now consider a 3-bit input gate that is universal *provided we can fix some input bits and ignore some output bits.* This is the **Toffoli gate**, a generalization of the controlled-NOT gate:

$$T(x, y, z) = (x, y, z \oplus xy).$$

Note that $T(T(x,y,z)) = (x,y,z)$, *i.e.*, T is its own inverse. We can generalize this gate to the n-input Toffoli gate,

$$T_n(x_1, x_2, \ldots, x_{n-1}, y) = (x_1, x_2, \ldots, x_{n-1}, y')$$

where $y' = y \oplus x_1 x_2, \ldots, x_{n-1}$. In other words, the last bit is complemented provided the first $n-1$ bits are 1. The Toffoli gate $T = T_3$ and controlled XOR gate T_2 are thus special cases. We construct T_4 from T_3 using the circuit in Figure 22.3(c). Note that there is an extra line that is set to 0, and whose value is preserved (the role of the third T-gate is to reset this bit to 0). In Figure 22.3(d), we use an extra gate and now the extra line can have any initial value. It is easy to generalize these constructions to obtain T_n.

Next, suppose $x, y \in \mathbb{B}^n$ and their Hamming distance is 1. For each i, ith bit of x is denoted x_i. We can use T_n to compute the transposition $x \leftrightarrow y$ as follows: if the ith bit of x and y are different (here i is unique since their Hamming distance is 1), then we can use T_n to complement this bit iff the other bits are precisely the bits in x (and hence y). This is easily done: for each $j \neq i$, if $x_j = y_j = 0$, then we complement the jth input before it enters T_n, and complement the jth output bit again, just after it exits T_n. Now, negation is easily implemented using $NOT(z) = T(1,1,z)$.

In general, when the Hamming distance between x and y is $d \geq 1$, we simply use a composition of d transpositions of the kind described in the previous paragraph. Finally, an arbitrary permutation of $\mathbb{B}^n$ is reduced to a composition of transposition. This proves:

Theorem 1 *All Boolean functions can be simulated on a reversible circuit based on the Toffoli gate T, provided we can introduce additional input and output lines.*

As we saw, the additional input lines can be arbitrary (but if we can preset its values, the size of the circuit can be smaller).

22.1.5 Bennett's Scheme for Reversible Computation

Bennett's basic result [7] says that *any computable function $F(x) = y$ can be converted into a reversibly computable function of the form $\widetilde{F}(x) = (x, y)$. Moreover, if F is computing in time T then $\widetilde{F}$ can be computed in time $O(T)$.* We will prove this.

The basic idea is simply to keep a history of the computational steps in performing F. Begin with any machine M that computes some function

$$M(input_n) = (output_m).$$

The subscript n, m tells us how many bits are in the input and output arguments. For the present discussion, assume that M is actually a Boolean circuit. But it will be clear that the method is general.

We first replace M by a reversible circuit M_1. For instance, if M uses only AND and OR gates only, we can make them reversible by adding extra lines. This reversible circuit M_1 can be further replaced by Toffoli gates if desired. Also, each original input line in M must be duplicated, as many times as it is used in the Boolean circuit. The additional input lines must be preset, and the additional output lines will in general contain garbage. Actually, we can think of the garbage as a record of our computation. Let these extra lines be called $record_k$ (there are k lines). It is possible to stop at this point and say that we have achieved our objectives.

However, let us impose an additional requirement: all the extra k output lines must be reset to 0. This requirement will be useful later, if we want to use implement our scheme by quantum computers later: resetting $record_k$ to 0 is needed to to avoid interference when we read the desired $output_m$. We proceed as follows: add another m extra input lines preset to 0 – call these inputs $zeros_m$. They will hold our eventual output. We proceed to simulate M in a reversible manner using these extra input lines. When $output_m$ is produced, we make an extra copy of $output_m$ in extra lines $zeros_m$. Recall that we can copy values using the T_2 gates, as in (4). Finally, we run the reversible circuit M_1 backwards, with $output_m$ and the extra output lines as input! Eventually, these will revert to $input_n, zero_k$ (where $zero_k$ is the original value of the $record_k$). Thus, we have computed the function

$$(input_n, zeros_k, zeros_m) \mapsto (input_n, zeros_k, output_m).$$

If we view the scratch space $zeros_k$ as internal to the machine, we have the desired function

$$\widetilde{F} : (input_n, zeros_m) \mapsto (input_n, output_m).$$

Shor noted that Lecerf [19] obtained a result similar to Bennett's. Basically, any function which outputs a copy of its input (in addition to other outputs) can be made reversible. Note that although time is not increased in this scheme, space may be as bad as $\Omega(T)$. It is possible to improve the space utilization by trading it for increased time [9].

Exercise

Exercise 22.1.1: (i) Prove that if A, B are linear transformations on $S = \mathbb{C}^m$ and $T = \mathbb{C}^n$ then $A \otimes B$ is a linear transformation on $S \otimes T$.
(ii) Prove the same result when the vectors in S, T are restricted to unit length. □

Exercise 22.1.2: Consider reversible circuits to compute the following transformation: $f : \mathbb{B}^{2n} \to \mathbb{B}^{2n}$ where $f(x, y) = (x, y^2)$ where $0 \le x, y < 2^n$ are n-bit binary numbers and y^2 is taken modulo 2^n. Construct the circuit explicitly for the case $n = 3$ using the family of T_n gates. □

Exercise 22.1.3: Why do we need the two extra bits when we want to convert $f : \mathbb{B}^n \to \mathbb{B}^m$ into a permutation? For instance, let $f'' : \mathbb{B}^{m+n} \to \mathbb{B}^{m+n}$ where $f''(x, 0^m) = (x, f(x))$. Can we complete this specification of f'' into an invertible function? □

Exercise 22.1.4: Design a full adder using T_2 and T_3 gates. This circuit has 4 lines a, b, c, d where a, b, c are the two input bits plus a carry-in bit, and d is set to 0. The output lines are a', b', c', d' where c' is the sum and d' the carry-out. Consider a', b' to be garbage. Note: 4 gates suffices. □

Exercise 22.1.5: Show by a direct argument that T_3 cannot be constructed out of circuits involving T_2 and other two-input gates. □

Exercise 22.1.6: We will be computing in $\mathbb{Z}_2$ (modulo 2) in the following.
(i) Enumerate the 6 invertible 2×2 matrices.
(ii) Show that all invertible 2-input functions are linear: $(x, y) \mapsto (x, y)M + (a, b)$ where M is an invertible matrix from (i) and a, b are constants.
(iii) Show that any circuit composed of linear gates is linear.
(iv) Conclude that the set of reversible 2-inputs and 1-input gates is not universal.
(v) Give an nonlinear invertible function $f : \mathbb{B}^3 \to \mathbb{B}^3$. Show that it is nonlinear as well as give an implementation of f by Toffoli gates. □

Exercise 22.1.7:
(i) Show the recursive construction of T_n from T in which we use only one extra bit of scratch space. What is the number of T-gates used? (ii) Show that with 3 scratch bits, initiallized to 0, we can construct T_n with $2n - 5$ T-gates. □

Exercise 22.1.8: Show that any 4-input circuit composed from $T(x, y, z)$ must compute an even permutation of $\mathbb{B}^4$. Conclude that such a circuit cannot compute T_4. □

End Exercise

22.2 Quantum Circuits

In quantum circuits, unlike classical circuits, we have qubits instead of bits on each wire or line. Next, we must use quantum gates, which is a generalization of reversible gates. Each quantum gate is a unitary transformation on a fixed number of qubits. These unitary transformations are chosen from some finite basis. At the end of the computation, a measurement is made of some of the qubits. This "measurement at the end" is a canonical choice; in practice, one may wish to make measurements throughout the computation. Quantum circuits were introduced by David Deutsch in 1981.

A gate or function that manipulates n qubits is represented by a $2^n \times 2^n$ unitary matrix. The following simple unitary matrix H operates on 1 qubit:

$$H := \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{6}$$

It is easy to check that the matrix[5] H is unitary: $H^*H = I$. We identify H with the unitary transformation that transforms the state $|0\rangle$ and $|1\rangle$ as follows:

$$\begin{aligned} H : |0\rangle &\mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |1\rangle &\mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle, \end{aligned}$$

[5]The letter H refers to Hadamard. Usually, Hadamard matrices are non-singular matrices with entries with ± 1 entries. Here we need to scale the ± 1 entries by the factor $1/\sqrt{2}$.

In terms of matrix multiplication, $|0\rangle$ and $|1\rangle$ are viewed as the column vectors $e_0 = (1,0)^T$ and $e_1 = (0,1)^T$, respectively, and the transformation is just matrix multiplication by H.

Recall that in the Stern-Gerlach device figure 22.1, we could change the basis of the measurement by just rotating the apparatus along the axis of the electron path. If we rotate the apparatus by 90°, then this is essentially the same as using the basis $H|0\rangle$ and $H|1\rangle$.

We can generalize H to

$$H_n := \underbrace{H \otimes H \otimes \cdots \otimes H}_{n}$$

which is a $2^n \times 2^n$ matrix. This is now a unitary operator on n-quwords. For instance,

$$H_2 = \frac{1}{2}\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right]. \tag{7}$$

This transformation will be useful in various quantum algorithms, as it prepares all possible 2^n states for a subsequent computation.

Convention: The operations of a $2^n \times 2^n$ unitary matrix M on states of n-quwords uses a natural convention which may be worth spelling out. The pure states are ordered lexicographically, in the order $|0^n\rangle, |0^{n-1}1\rangle, \ldots, |1^n\rangle$ or equivalently,

$$|0\rangle, |1\rangle, |2\rangle, \ldots, |2^n - 1\rangle.$$

This ordering is used to label the rows and columns of M. For $i = 0, \ldots, 2^n - 1$, the elementary 2^n-vector e_i which has 1 in the ith position and 0 elsewhere represents the pure state $|i\rangle$. The state s of the quword is therefore a 2^n-vector of length 1. Then the transformation M operates on state s by matrix-vector multiplication, Ms For instance,

$$H_n|\underbrace{00\cdots 0}_{n}\rangle = \frac{1}{2^{n/2}} \sum_{i=0}^{2^n-1} |i\rangle.$$

22.2.1 The Unitary Matrices of Reversible Circuits

We had said that that reversible circuits are special cases of quantum circuits. This is actually easy to show, and instructive. We use the fact that the Toffoli gate $T(x,y,z) = T_2(x,y,z)$ is universal for reversible circuits. We must now interpret the gate $T(x,y,z)$ as a unitary 8×8 matrix U_3 operating on linear combinations of the pure states $|xyz\rangle = |x\rangle \otimes |y\rangle \otimes |z\rangle$. We have

$$U_3(|x\rangle \otimes |y\rangle \otimes |z\rangle) = U_3(|xyz\rangle) = \begin{cases} |xyz\rangle) & \text{if } x = 0 \text{ or } y = 0, \\ |111\rangle & \text{if } x = y = 1, z = 0, \\ |110\rangle & \text{if } x = y = z = 1. \end{cases}$$

With the usual labeling conventions, U_3 becomes the following matrix

$$U_3 = \left[\begin{array}{cccccc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] \tag{8}$$

It is easy to check that this is unitary. When we compose gates into circuits, we are building unitary matrices from the unitary matrices represented by the individual gates in a natural way: For instance, if $|x\rangle = |x_1x_2x_3\rangle$ is the state of 3 qubits that are being operated on by T_3, and $|y\rangle = |x_4 \cdots x_n\rangle$ is the state of the rest of the qubits, then the overall unitary matrix corresponding to this transformation is the tensor product

$$V_3 = U_3 \otimes I$$

where I is a $2^{n-3} \times 2^{n-3}$ identity matrix. Such matrices V_3 are **permutation matrices**, namely, each row and each column has exactly one 1, with the rest of the entries 0. The universality of T_3 implies that every permutation matrix can written as a product of matrices such as V_3 (we need to generalize V_3 to allow any permutations of the n qubits).

Although classically, U_3 is meant to act on pure states, under the quantum interpretation, the vectors that U_3 now acts on is now allowed to be any unit vector (supposition of pure states). Deutsch showed the existence of a universal quantum gate $D(x, y, z)$. It is a generalization of Toffoli's gate: if $x, y, z \in \mathbb{B}$ then $D(|x, y, z\rangle) = |x, y, z\rangle$ if both $x \wedge y \neq 1$; otherwise $D(x, y, z) = |x, y, U(z)\rangle$ where U is a 2×2 unitary transformation.

22.2.2 Structure of 2×2 Unitary Matrices.

The following set of three 2×2 unitary matrices are the **Pauli matrices**:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \sigma_y = \begin{bmatrix} 0 & -\mathbf{i} \\ \mathbf{i} & 0 \end{bmatrix}, \qquad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{9}$$

Sometimes they are also denoted $\sigma_1, \sigma_2, \sigma_3$, respectively. Also, σ_0 can be taken as the identity matrix. Our goal now characterize all unitary 2×2 matrices. following Barenco et al [1].

Let $U = [x|y]$ where $x, y \in \mathbb{C}^2$ are column vectors. Suppose $x = (re^{\mathbf{i}\theta}, r'e^{\mathbf{i}\theta'})^T$ where r, θ, r', θ' are real and $\mathbf{i} = \sqrt{-1}$. Then $|x|^2 = 1$ implies $r^2 + r'^2 = 1$ and hence we may write $x = (\cos\alpha e^{\mathbf{i}\theta}, \sin\alpha e^{\mathbf{i}\theta'})^T$ for some real α. Similarly, let $y = (\sin\beta e^{\mathbf{i}\psi}, \cos\beta e^{\mathbf{i}\psi'})^T$ for some real β, ψ, ψ'. Next, $x^* y = 0$ implies $\cos\alpha e^{-\mathbf{i}\theta} \sin\beta e^{\mathbf{i}\psi} + \sin\alpha e^{-\mathbf{i}\theta'} \cos\beta e^{\mathbf{i}\psi'} = 0$, or in matrix notation,

$$\begin{bmatrix} \cos(\psi - \theta) & \cos(\psi' - \theta') \\ \sin(\psi - \theta) & \sin(\psi' - \theta') \end{bmatrix} \begin{pmatrix} \cos\alpha\sin\beta \\ \sin\alpha\cos\beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Thus the determinant of the 2×2 matrix vanishes,

$$\cos(\psi - \theta)\sin(\psi' - \theta') - \sin(\psi - \theta)\cos(\psi' - \theta') = 0.$$

Writing $2\delta = \theta' - \theta$ and $2\delta' = \psi' - \psi$, this gives $\sin(2\delta' - 2\delta) = 0$. Thus $\delta = \delta'$ and we have

$$x = e^{\mathbf{i}(\theta+\delta)}(\cos\alpha e^{-\mathbf{i}\delta}, \sin\alpha e^{\mathbf{i}\delta})^T, \qquad y = e^{\mathbf{i}(\psi+\delta)}(\sin\beta e^{-\mathbf{i}\delta}, \cos\beta e^{\mathbf{i}\delta})^T.$$

In matrix notation, we may write $U = ABC$ where A, B, C are the matrices

$$A = \begin{bmatrix} e^{-\mathbf{i}\delta} & 0 \\ 0 & e^{\mathbf{i}\delta} \end{bmatrix}, \qquad B = \begin{bmatrix} \cos\alpha & \sin\beta \\ \sin\alpha & \cos\beta \end{bmatrix}, \qquad C = \begin{bmatrix} e^{\mathbf{i}(\theta+\delta)} & 0 \\ 0 & e^{\mathbf{i}(\psi+\delta)} \end{bmatrix}.$$

Since A, C are unitary, it follows that B must be unitary. This means $BB^* = I$ and so $\cos\alpha\sin\alpha + \cos\beta\sin\beta = 0$. This implies $\alpha = -\beta$.

$$B = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}.$$

One more freedom can be restricted when we assume that U is special, *i.e.*, $\det(U) = 1$. Since $\det(A) = \det(B) = 1$, we have $\det(C) = 1$. This means $e^{\mathbf{i}(\theta+\psi+2\delta)} = 1$ or $\theta + \delta = -(\psi + \delta)$. By renaming $\theta + \delta$ to θ and $\psi + \delta$ to ψ, we obtain the result of Bloch:

THEOREM 2 *Every unitary matrix $U \in \mathbb{C}^{2\times 2}$ has the form*

$$U = \begin{bmatrix} e^{-\mathbf{i}\delta} & 0 \\ 0 & e^{\mathbf{i}\delta} \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} e^{\mathbf{i}(\theta)} & 0 \\ 0 & e^{\mathbf{i}(\psi)} \end{bmatrix}.$$

If U is special, then $\psi = -\theta$.

Toffoli-type quantum gates. Let $U = \begin{bmatrix} a_0 & b_0 \\ a_1 & b_1 \end{bmatrix}$ be any unitary matrix. Following [1], for any $n \geq 1$, we define the $2^n \times 2^n$ matrix

$$T_n(U) = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & a_0 & b_0 \\ & & & & a_1 & b_1 \end{bmatrix}.$$

This is clearly unitary. This generalizes the Tofolli gate $T_n(x_1, \ldots, x_{n-1}, y)$: T_n is just $T_n(U_\sigma)$ where $U_\sigma = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. We can similarly describe the action of T_n on the pure state $|x_1 \cdots x_n\rangle$ as follows: $T_n(U)(|x_1 \cdots x_n\rangle) = |y_1 \cdots y_n\rangle$ where $y_i = x_i$ for $i = 1, \ldots, n-1$ and

$$y_n = \begin{cases} x_n & \text{if some } x_i = 0 \text{ for some } i = 1, \ldots, n-1 \\ a_{y_n}|x_1 \cdots x_{n-1}0\rangle + b_{y_n}|x_1 \cdots x_{n-1}1\rangle & \text{if } x_i = 1 \text{ for all } i = 1, \ldots, n-1 \end{cases}$$

Thus the last qubit is transformed according to U iff the first $n-1$ qubits are 1. For instance,

$$T(|10\rangle) = a_0|10\rangle + a_1|11\rangle, \quad T(|11\rangle) = b_0|10\rangle + b_1|11\rangle.$$

The diagramatic representation of such gates is shown in figure 22.5.

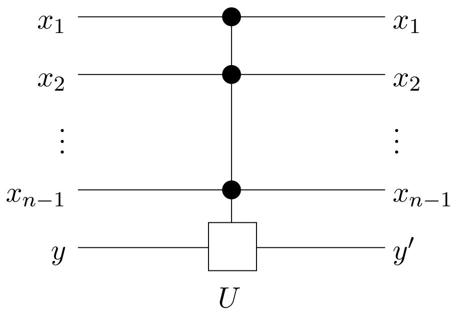


Figure 22.5: Toffoli-type quantum gates $T_n(U)$.

REMARK: While we have noted that reduction of quantum circuits to reversible circuits is a useful procedure, the results in this section also show quantum circuits are more powerful: for instance, there are no 2-input universal reversible gate while there are universal 2-input quantum gates. Also, no work-bits are needed in quantum circuits, in contrast to some reversible circuits.

22.2.3 No Cloning Theorem

Before we leave this topic of quantum circuits, we must prove a simple but fundamental result about unitary transformations from Wootters and Zurek (1982). This result says that we cannot copy an unknown quantum state perfectly. But first, recall the controlled XOR gate, $T_2(x, y)$. We have

$$T_2(x, 0) = (x, x).$$

Thus, the bit x has been copied or cloned. The next result shows that this is impossible with a quantum gate.

THEOREM 3 (NO CLONING) *Let $H = H_1 \otimes H_1$ be a Hilbert space where H_1 is of dimension at least 2. There does not exist a unitary transformation U such that for all $x \in H_1$,*

$$U(|x, 0\rangle) = |x, x\rangle$$

Proof. Let $|x\rangle \neq |y\rangle$ be two orthogonal states of H_1. By way of contradiction, suppose U exists. Then $U(|x, 0\rangle) = |x, x\rangle$ and $U(|y, 0\rangle) = |y, y\rangle$. If $z = \frac{1}{\sqrt{2}}(|x\rangle + |y\rangle)$, then

$$U(|z, 0\rangle) = \frac{1}{\sqrt{2}}(|x, x\rangle + |y, y\rangle)$$

by linearity of U. However, we note that

$$|z,z\rangle = |z\rangle \otimes |z\rangle = \frac{1}{2}(|x\rangle + |y\rangle) \times (|x\rangle + |y\rangle) = \frac{1}{2}(|x,x\rangle + |y,x\rangle + |y,x\rangle + |y,y\rangle).$$

This is contradiction because $U(|z,0\rangle) \neq |z,z\rangle$. **Q.E.D.**

One consequence of this theorem is in quantum cryptography: it implies that secure quantum key generation is possible. However, the no-cloning theorem says nothing about the possibility of good but imperfect copying. Indeed such "weak copying" techniques have been proposed.

Exercise

Exercise 22.2.1: Consider the Pauli matrices σ_i ($i = x, y, z$ or $i = 1, 2, 3$). Show
(i) $\sigma_i^2 = I$.
(ii) The eigenvalues of σ_i are ± 1.
(iii) $\sigma_{i-1}\sigma_i = \mathbf{i}\sigma_{i+1}$, where subscript addition is modulo 3. □

Exercise 22.2.2: Consider the definition of e^X as the series $\sum_{k\geq 0} X^k$. This definition can be extended to the case where X is a square matrix. Prove that $e^{\mathbf{i}\delta\sigma_z} = \begin{bmatrix} e^{\mathbf{i}\delta} & 0 \\ 0 & e^{-\mathbf{i}\delta} \end{bmatrix}$ where $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. □

Exercise 22.2.3: A $2^n \times 2^n$ unitary matrix U represents a classical gate iff it is 0/1 and every row and every column has exactly one 1. □

Exercise 22.2.4: Prove that $T_2(U)$ is universal for "most" unitary 2×2 U. □

End Exercise

22.3 Quantum Algorithms

There is a basic intuition that "quantum computers are inherently more powerful than classical computers". This can be formalized as questions about inclusion of complexity classes. Unfortunately, we are unlikely to resolve such questions directly because it would imply the resolution of some well-known conjectures in classical complexity theory. A more modest goal is to demonstrate individual problems that could be solved more efficiently using quantum computers than with classical computers. Even here, most positive results must be qualified in the sense that a quantum computer can solve a problem more efficiently than any *known* classical algorithm. We should also remember that quantum algorithms are inherently probabilistic, and hence we should only compare them to classical randomized algorithms.

But what kinds of problems can exploit the special capabilities of quantum computers? There is an obvious candidate task where quantum computers can do more efficiently than classical computers: the simulation of quantum computations! This was noted by Feynman. But what else? Let us note that the apparent advantage of quantum computers is the ability to simultaneously maintain (exponentially) many possible states. Hence the obvious way to exploit this is to evolve these states simultaneously. On the other hand, if we measure the quantum system, we only get one of these states (possibly with exponentially small probability). So in order for useful computation to be carried out, we need to have these states interfere in some controlled manner, to bias the probabilities towards the desirable states. One view of quantum computation (see Cleve et al [13]) is that it is basically an application of interferometry to multiparticle systems. This is so different from classical computers that completely new computational techniques must be developed before we can exploit its power.

22.3.1 Deutsch's Problem

Deutsch [14] gave a simple demonstration of the power of quantum computers over classical computers. Assume we have a quantum black box B that computes the 2-qubit transformation

$$B : (x, y) \mapsto B(x, y) = (x, y \oplus f(x))$$

for some unknown function $f : \mathbb{B}^1 \to \mathbb{B}^1$. There are only 4 possibilities for f: constant 0 function ($f(0) = f(1) = 0$), constant 1 function ($f(0) = f(1) = 1$), identity function ($f(x) = x$), or negation function ($f(x) = 1 - x$). Deutsch's problem is to determine whether $f(0) = f(1)$ (f is constant) or $f(0) \neq f(1)$ (f is balanced). If this were a classical tranformation, we would need to access the black box twice, with $x = 0$ and $x = 1$. But being quantum, we now show a one-access solution.

Consider the unitary matrix H in (6). If we first prepare our input to be $|0\rangle \otimes |1\rangle$, then subject it to the transformation $H_2 = H \otimes H$ (see (7)), we obtain

$$|x\rangle \otimes |y\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

If we pass this through our black box, the overall state becomes

$$\begin{aligned} B(|x\rangle \otimes |y\rangle) &= |x\rangle \otimes |y \oplus f(x)\rangle \\ &= |x\rangle \otimes \frac{1}{\sqrt{2}}(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= |x\rangle \otimes \frac{1}{\sqrt{2}}(-1)^{f(x)}(|0\rangle - |1\rangle) \\ &= (-1)^{f(x)}|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= (-1)^{f(x)}|x\rangle \otimes |y\rangle \end{aligned}$$

The first qubit is quite interesting:

$$\begin{aligned} (-1)^{f(x)}|x\rangle &= \frac{1}{\sqrt{2}}(-1)^{f(x)}(|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle) \\ &= (-1)^{f(0)}\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{f(1)+f(0)}|1\rangle) \\ &= \begin{cases} (-1)^{f(0)}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & \text{if } f(x) \text{ is constant} \\ (-1)^{f(0)}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \text{if } f(x) \text{ is balanced} \end{cases} \\ &= \begin{cases} (-1)^{f(0)}|x\rangle & \text{if } f(x) \text{ is constant} \\ (-1)^{f(0)}|y\rangle & \text{if } f(x) \text{ is balanced} \end{cases} \end{aligned}$$

Thus, if we measure this qubit using the basis $(|x\rangle, |y\rangle)$, we obtain $|x\rangle$ with probability 1 if $f(x)$ is constant, and $|y\rangle$ with probability 1 if $f(x)$ is balanced. Note that the $(-1)^{f(0)}$ is just phase information which does not affect the probability. Alternatively, we can apply H to the first qubit again (recall that $H^{-1} = H$), the possible states are $|0\rangle$ or $|1\rangle$ depending on the nature of f. Now we measure in the original basis. The quantum circuit to perform this computation is shown in figure 22.6(a). We have just described the improved solution of Cleve et al [13].

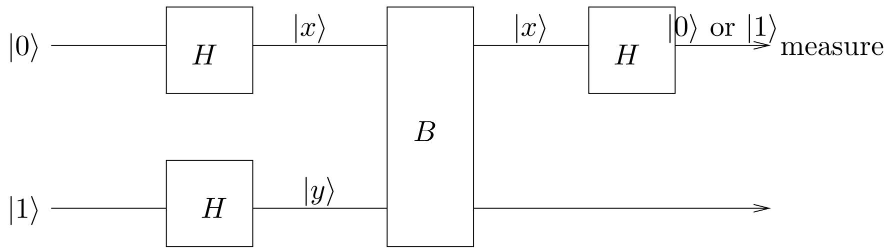
Generalization. Deutsch and Jozsa [16] extended the above example to allow arbitrarily large input sizes. This Deutsch-Jozsa problem is as follows. Suppose $f : \mathbb{B}^n \to \mathbb{B}$ is an unknown function, but it is either a constant function ($f(x)$ is always 0 or always 1) or it is balanced (the number of times $f(x) = 0$ is to 2^{n-1}). We need to decide whether f is constant or balanced. We are given a quantum blackbox B_n which takes a $(n+1)$-quword as input and which applies the function f as follows:

$$B_n(|x_1, \ldots, x_n y\rangle) = |x_1, \ldots, x_n\rangle \otimes |y \oplus f(x_1, \ldots, x_n)\rangle.$$

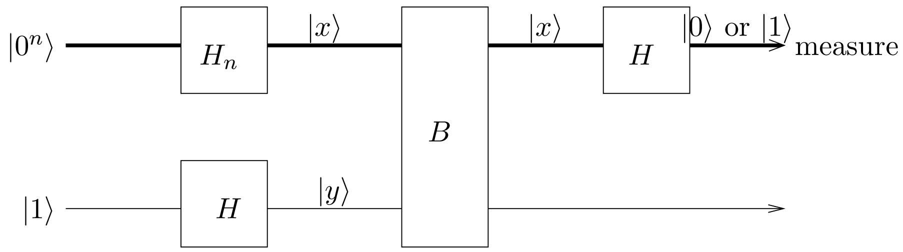
Using a deterministic algorithm, it seems that we need to make at least $2^{n-1} + 1$ evaluations of B_n, in the worst case. A randomized procedure can do better (Exercise). But we now show that a single call to the black-box is sufficient using a quantum transformation.

The method is a straightforward generalization of the original solution. We prepare the input to be $|0^n 1\rangle$ and apply $H_{n+1} = H \otimes \cdots \otimes H$ to this input. This produces the state

$$\left(\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{10}$$



(a) Simple



(a) General case

Figure 22.6: Deutsch's Problem: (a) simple case, (b) general case

Then we pass this state through B_n. Suppose $y = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and $x = x_1, \ldots, x_n$ is a pure state of the first n bits. Then

$$B_n(|x_1, \ldots, x_n y\rangle) = |x\rangle \otimes \frac{1}{\sqrt{2}}(|f(x) \oplus 0\rangle - |f(x) \oplus 1\rangle) = |x\rangle \otimes \frac{1}{\sqrt{2}}(-1)^{f(x)}(|0\rangle - |1\rangle).$$

Hence the overall state of the quword, after passing the state (10) through B_n, is

$$\left(\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|x\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Finally, apply H_{n+1} again to get

$$\left(\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} \left(\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} ((-1)^{f(x)}(-1)^{x \cdot y}|y\rangle)\right)\right) \otimes H(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle))$$

But

$$\left(\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} \left(\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} ((-1)^{f(x)}(-1)^{x \cdot y}|y\rangle)\right)\right) \otimes H(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle))$$

If we now measure the first n qubits. if f is constant, then a measurement of the first $n-1$ qubits will give $|0^{n-1}\rangle$ with probability 1; if f is balanced, the same measurement would never give the state $|0^{n-1}\rangle$.

22.3.2 Quantum Discrete Fourier Transform

We introduce the quantum analogue of the well-known Discrete Fourier Transform (DFT). The quantum version of this (QFT) is obtained as a natural adaptation. We derive a quantum circuit for computing QFT. This algorithm will be used later for integer factorization.

Let $N \in \mathbb{N}$, and $x = (x_0, \ldots, x_{N-1})^T \in \mathbb{C}^N$. We define the **discrete Fourier transform** of x to be $DFT(x) = Fx$ (a matrix vector product), where F is the following $N \times N$ matrix

$$F = F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ & \vdots & & & \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix} \tag{11}$$

and $\omega = e^{\mathbf{i}2\pi/N}$. Note that F is symmetric. We now prove that $F = F_N$ is unitary. Two preliminary remarks are useful: In general, the conjugate of $e^{\mathbf{i}\theta}$ is $\overline{e^{\mathbf{i}\theta}}$, and thus $\overline{\omega} = e^{-\mathbf{i}2\pi/N}$. Second, note that ω is an Nth **root of unity** meaning that $\omega^N = 1$. But more is true: it is a **primitive root of unity**, meaning that $\omega^n = 1$ implies n is a multiple of N. This amounts to showing that $F^*F = I$, or the (i,j)th entry of F^*F is 1 iff $i = j$:

$$\begin{aligned} (F^*F)_{ij} &= \frac{1}{N}\sum_{k=0}^{N-1} \overline{\omega^{ki}}\omega^{kj} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} \omega^{-ki}\omega^{kj} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} \omega^{k(j-i)} \\ &= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \end{aligned}$$

The last equation is clearly true when $i = j$, since the sum is N in this case. When $i \neq j$, the sum $\sum_{k=0}^{N-1} \omega^{k(j-i)}$ vanishes because of the simple identity

$$(1-x)\sum_{k=0}^{N-1} x^k = 1 - x^N. \tag{12}$$

Plugging $x = \omega^{j-i}$, we conclude that the righthand side is 0 (since ω is a N-th root of unity). But $1-x = 1-\omega^{j-i} \neq 0$ (since ω is a primitive N-th root of unity). This implies that the sum $\sum_{k=0}^{N-1} x^k$ must vanish.

Quantum Circuit for QFT. Since $DFT(x)$ is a unitary transformation, we should be able to compute it with a quantum circuit. The circuit turns out to be fairly simple, but to verify its correctness, we need some preparatory development.

We now choose $N = 2^n$. Previously, we wrote $\omega = e^{\mathbf{i}2\pi/N}$; now we slightly modify the notation and write $\omega_\ell = e^{\mathbf{i}2\pi/2^\ell}$ for all $\ell \in \mathbb{N}$. For instance, $\omega_0 = 1$ and $\omega_1 = e^{\mathbf{i}\pi} = -1$.

Let $|x\rangle = \sum_{k=0}^{N-1} x_k|k\rangle$. We define the **quantum Fourier transform (QFT)** as follows:

$$QFT(|x\rangle) = |y\rangle = \sum_{k=0}^{N-1} y_k|k\rangle \tag{13}$$

where

$$y_k = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j\omega_n^{jk}. \tag{14}$$

We give a simple form for $|y\rangle$ in case $|x\rangle = |j\rangle$ is a pure state, where $j = (j_1, \ldots, j_n)_2$ in binary. Then (14) becomes

$$y_k = \frac{1}{\sqrt{N}}\omega_n^{jk} \tag{15}$$

since $x_i = 1$ if $i = j$, and $x_i = 0$ otherwise. and so (13) becomes

$$QFT(|j\rangle) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} \omega_n^{jk}|k\rangle \tag{16}$$

Lemma 4

$$QFT(|j_1 \cdots j_n\rangle) = \frac{1}{\sqrt{N}} \bigotimes_{\ell=1}^{n} (|0\rangle + \omega_\ell^j |1\rangle) \tag{17}$$

$$= \frac{1}{\sqrt{N}} (|0\rangle + \omega_1^j |1\rangle) \otimes (|0\rangle + \omega_2^j |1\rangle) \otimes \cdots \otimes (|0\rangle + \omega_n^j |1\rangle) \tag{18}$$

Proof. Write $|j\rangle = |j_1, \ldots, j_n\rangle = |j_1\rangle \otimes \cdots \otimes |j_n\rangle$.

$$\begin{aligned}
QFT(|j\rangle) &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_n^{jk} |k\rangle, \qquad \text{(from (13), (15))} \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0} \omega_n^{j(\sum_{i=0}^{N-1} k_i 2^{n-i})} |k_1, \ldots, k_n\rangle, \qquad \text{(rewriting the sum over } k) \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0} \bigotimes_{\ell=1}^{n} \omega_n^{jk_\ell 2^{n-\ell}} |k_\ell\rangle, \qquad \text{(rewriting a tensor)} \\
&= \frac{1}{\sqrt{N}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0} \bigotimes_{\ell=1}^{n} \omega_\ell^{jk_\ell} |k_\ell\rangle, \qquad \text{(by definition of } \omega_\ell) \\
&= \frac{1}{\sqrt{N}} (|0\rangle + \omega_1^j |1\rangle) \otimes (|0\rangle + \omega_2^j |1\rangle) \otimes \cdots \otimes (|0\rangle + \omega_n^j |1\rangle)
\end{aligned}$$

Note that this agrees with (18). **Q.E.D.**

We have two remarks:

1. The notation in (17), although essentially equivalent to (18), can be ambiguous unless we understand the convention that a tensor product of the form $\otimes_{\ell=1}^n$ has to be taken in order of increasing ℓ. This remark may become clearer when we describe the quantum circuit to compute QFT.
2. The coefficient ω_ℓ^j in this lemma needs to be decoded:

$$\begin{aligned}
\omega_\ell^j &= e^{\mathbf{i}2\pi 2^{-\ell}(\sum_{i=1}^{n} j_i 2^{n-i})} \\
&= e^{\mathbf{i}2\pi(\sum_{i=1}^{\ell} j_{n+i-\ell} 2^{-i})}, \qquad \text{(keeping only the fractional part of the exponent of } e^{\mathbf{i}2\pi}. \\
&= e^{\mathbf{i}2\pi(0.j_{n+1-\ell} j_{n+2-\ell} \cdots j_n)}
\end{aligned}$$

where the $(0.j_{n+1-\ell} j_{n+2-\ell} \cdots j_n)$ is a binary rational. In other words, although the notation "ω_ℓ^j" suggests that this coefficient depends on j, it only depends on the last ℓ bits of $j_1, j_2, \ldots, j_n$. In the following, we shall write

$$\omega^{(0.j_1 j_2 \cdots j_\ell)} \tag{19}$$

for $e^{\mathbf{i}2\pi(0.j_1 j_2 \cdots j_\ell)}$.

One Stage of QFT Circuit. The QFT circuit will consist of n stages. It is sufficient to understand a single stage of this process. In fact the first stage is representative of all the other stages, and is illustrated in Figure 22.7.

The first gate is represented by the standard H-matrix. This H-gate transforms a pure qubit state $|j_1\rangle$ as follows: $H(|0\rangle) = (|0\rangle + |1\rangle)/\sqrt{2}$ if $j_1 = 0$ and $H(|1\rangle) = (|0\rangle - |1\rangle)/\sqrt{2}$ if $j_1 = 1$. We summarize this by writing

$$H(|j_1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + \omega_1^{j_1} |1\rangle). \tag{20}$$

Viewing the first H-gate as a transformation of all the qubits of $|j\rangle$, we obtain

$$H(|j\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + \omega_1^{j_1} |1\rangle) \otimes |j_2 \cdots j_n\rangle.$$

If $n = 1$, we are done. If $n \geq 2$, we introduce $n - 1$ transformations of the following type. For $\ell \geq 2$, the transformation R_k is a 2-qubit gate that achieves a "controlled rotation", and is given by the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \omega_\ell & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

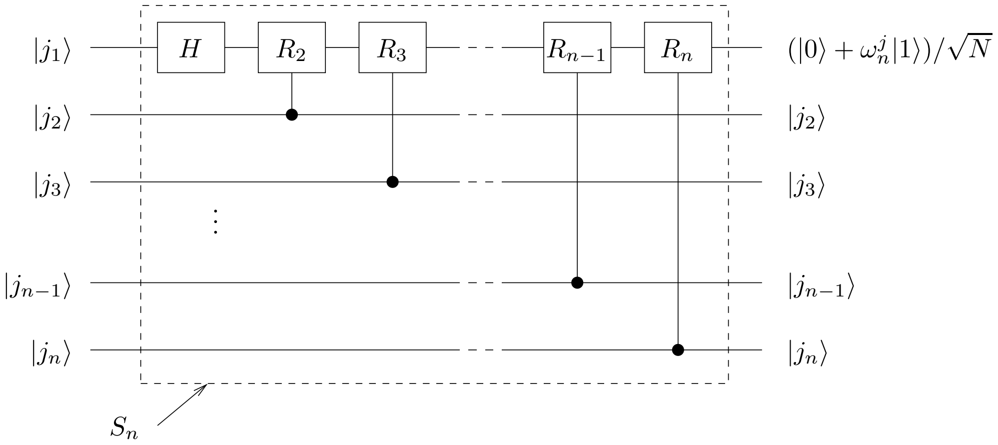


Figure 22.7: Stage One of the QFT circuit

In Figure 22.7, R_2 is applied to the first and second qubits of $|j_1 \cdots j_n\rangle$, where the second qubit j_2 acts as the control wire. Thus

$$\begin{aligned} R_2(|00\rangle) &= |00\rangle \\ R_2(|01\rangle) &= \omega_2|01\rangle \\ R_2(|10\rangle) &= |10\rangle \\ R_2(|11\rangle) &= |11\rangle \end{aligned}$$

The output of H on lines 1 and 2 is $\frac{1}{\sqrt{2}}(|0\rangle + \omega_1^{j_1}|1\rangle) \otimes |j_2\rangle$. Considering the case $j_2 = 0$ and $j_2 = 1$ separately, R_2 yields:

$$\begin{aligned} R_2((|0\rangle + \omega_1|1\rangle) \otimes |0\rangle) &= (|0\rangle + \omega_1|1\rangle) \otimes |0\rangle, \\ &= (|0\rangle + \omega^{(0.10)}|1\rangle) \otimes |0\rangle, \\ R_2((|0\rangle + \omega_1|1\rangle) \otimes |1\rangle) &= (|0\rangle + \omega_2\omega_1|1\rangle) \otimes |1\rangle, \\ &= (|0\rangle + \omega^{(0.11)}|1\rangle) \otimes |1\rangle, \end{aligned}$$

This proves that

$$\begin{aligned} R_2(H(|j_1 j_2\rangle)) &= \frac{1}{\sqrt{2}}(|0\rangle + \omega^{(0.j_1 j_2)}|1\rangle) \otimes |j_2\rangle) \\ &= \frac{1}{\sqrt{2}}(|0\rangle + \omega_2^{j_1 j_2}|1\rangle) \otimes |j_2\rangle). \end{aligned}$$

Continuing the same way, each R_ℓ-gate simply transforms the factor $(|0\rangle + \omega^{0.j_1 j_2 \cdots j_{\ell-1}}|1\rangle)$ into $(|0\rangle + \omega^{0.j_1 j_2 \cdots j_{\ell-1} j_\ell}|1\rangle)$ (by appending j_ℓ to the the binary fraction in the exponent). The final result is that line 1 has the value,

$$\frac{1}{\sqrt{2}}(|0\rangle + \omega^{(0.j_1 j_2 \cdots j_n)}|1\rangle).$$

By comparing this result to lemma 4, we see that this result should really appear in line n in the QFT circuit. But we will postpone this transposition until the end. Lines 2 to n have their original values unchanged.

Putting together the stages. Let S_n be the circuit represented by stage 1. It is now clear that we can continue this process on lines 2 to line n, but applying the circuit S_{n-1} instead. as a result, line 2 will have the value $(|0\rangle + \omega_{n-1}^j|1\rangle)/\sqrt{2}$, Finally, line n is transformed by S_1 which is just a single H-gate, yielding $(|0\rangle + \omega_1^j|1\rangle)/\sqrt{2}$ on this line. The resulting circuit is seen in Figure 22.8.

We are almost done, except the values in the output lines of Figure 22.8 are in reverse order (compare lemma 4). But we know that transposing any two bits (quantum or classical) can be achieved by three T_2 gates. We can therefore introduce $N/2$ such gates to exchange the outputs of line i and line $N - i - 1$. This completes the description of the QFT circuit.

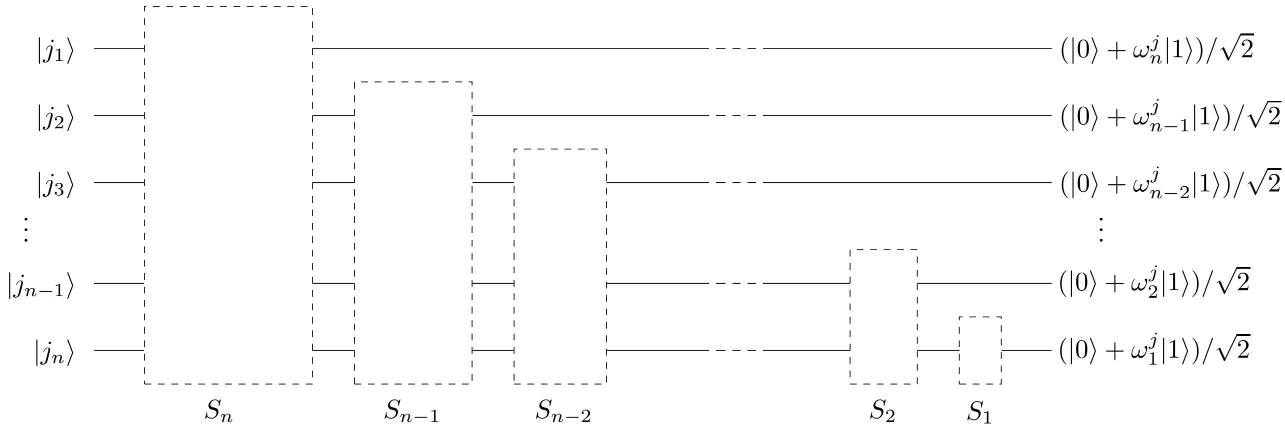


Figure 22.8: Quantum circuit for Reverse Fourier transform

Complexity Analysis. Since each stage uses $\Theta(n)$ gates, the overall number of gates is $\Theta(n^2)$.

At this point, we should pause and consider what has been accomplished: we can compute QFT using $O(n^2)$ quantum gates. This is exponentially smaller than any classical construction which surely need $\Omega(N) = \Omega(2^n)$ gates. But exploiting this result is hardly obvious. For one thing, it is unclear how to physically prepare an arbitrary quantum state $|x\rangle$. Even if we can do this, it is unclear how to extract the resulting values stored in $|y\rangle = QFT(|x\rangle)$.

22.3.3 Phase Estimation Problem

The phase estimation problem is one of the most basic tasks we might want to carry out. Suppose U is a unitary operator U, given as blackbox. This simply means that we can use it as a primitive gate in our quantum circuits. Furthermore, we are given an eigenvector $|v\rangle$ of U. If the associated eigenvalue is $\omega^\phi = e^{\mathbf{i}2\pi\phi}$, then

$$U|v\rangle = \omega^\phi|v\rangle.$$

Call ϕ the **phase angle**, and we may assume $0 \le \phi < 1$. The problem of **Phase Estimation** is, given U and $|v\rangle$ and also $n \in \mathbb{N}$ and $\varepsilon > 0$, to compute an approximation of ϕ to n bits of precision, with probability of failure $\le \varepsilon$. The approximation $\widetilde{\phi}$ has n bits of precision if $|\widetilde{\phi} - \phi| \le 2^{-n}$.

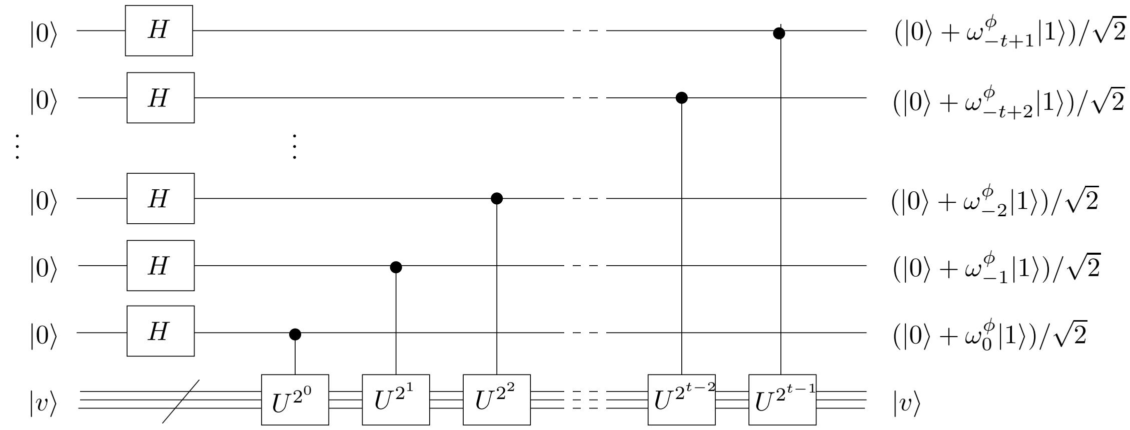


Figure 22.9: Stage 1 of Phase Estimation

There are two stages in phase estimation. The first stage is illustrated in Figure 22.9. Note that there are two registers (quwords):

(1) Register 1 comprise t qubits that are used to readout the final estimate of ϕ. At the end, we will measure this register 1, and this value, interpreted as a binary rational $0.j_1j_2\cdots j_t$ is regarded as the estimate of ϕ. The value

of t will be determined later, as it depends on the input parameters n and ε.
(2) Register 2 are used to carry the eigenvector $|v\rangle$.

Consider the result of applying U^{2^ℓ} $(\ell \geq 0)$ to the output of H:

$$\begin{aligned}
U^{2^\ell}(|0\rangle \otimes |v\rangle) &= |0\rangle \otimes |v\rangle \\
U^{2^\ell}(|1\rangle \otimes |v\rangle) &= \omega^{2^\ell \phi}|1\rangle \otimes |v\rangle \\
U^{2^\ell}((|0\rangle + |1\rangle) \otimes |v\rangle) &= (|0\rangle + \omega^{2^\ell \phi}|1\rangle) \otimes |v\rangle \\
&= (|0\rangle + \omega^{\phi}_{-\ell}|1\rangle) \otimes |v\rangle
\end{aligned}$$

This justifies the output on Register 1 as specified in the lines of Figure 22.9.

Phase Two. We motivate the construction of the second phase. Suppose

$$\phi = 0.\phi_1\phi_2\cdots\phi_t, \qquad \phi_i \in \{0,1\}.$$

Then we observe the output lines of Register 1 in Figure 22.9 is simply the outputs specified by Lemma 4. For instance, line one's output in Figure 22.9 is $(|0\rangle + \omega^{\phi}_{-t+1}|1\rangle)/\sqrt{2}$, which is equal to $(|0\rangle + \omega^{0.\phi_t}|1\rangle)/\sqrt{2}$. Line one of Lemma 4 has output $(|0\rangle + \omega^{j}_{1}|1\rangle)/\sqrt{2}$, which is equal to $(|0\rangle + \omega^{0.j_n}|1\rangle)/\sqrt{2}$. So the two outputs are the same once we identify n with t and $j_1, \ldots, j_n$ with $\phi_1, \ldots, \phi_t$.

This means, if we apply the inverse QFT circuit of the previous section as our second phase, we would obtain as output the pure state $|\phi_1 \cdots \phi_t\rangle$. This inverse QFT is indeed our second phase, even when ϕ is not a binary rational. But the proof that it yields a good estimate of ϕ in general is more subtle and will be taken up next.

Finally, the estimate of ϕ is obtained by measuring Register 1. This yields t bits which is interpreted as a binary rational $0.b_1b_2\cdots b_t$ $(b_i \in \{0,1\})$ and taken as the estimate of ϕ.

Error Analysis. Let $\phi < 1$ be an arbitrary real. Suppose $\widetilde{\phi} = 0.\phi_1 \cdots \phi_t$ is a binary rational such that

$$\delta := |\phi - \widetilde{\phi}| \leq 2^{-t-1}. \tag{21}$$

Ideally, we would like our final measurement to yield $|\phi_1, \ldots, \phi_t\rangle = |2^t\widetilde{\phi}\rangle$ in Register 1. But short of this, our current goal is to obtain pure state $|j\rangle = |j_1, \ldots, j_t\rangle$ such that

$$\Pr\{|\phi - 0.j_1, \ldots, j_t| < 2^{-n}\} \geq 1 - \varepsilon. \tag{22}$$

where n and ε are user specified parameters. We show how to choose t so that we obtain the guarantee (22).

The output of Phase 1 is

$$\frac{1}{2^{t/2}}(|0\rangle + \omega^{\phi}_{-t+1}|1\rangle) \otimes (|0\rangle + \omega^{\phi}_{-t+2}|1\rangle) \otimes \cdots \otimes (|0\rangle + \omega^{\phi}_{0}|1\rangle) = \frac{1}{2^{t/2}} \sum_{\ell=0}^{2^t-1} \omega^{\phi\ell}|\ell\rangle$$

The inverse QFT is basically the same as QFT, except that we replace ω by its conjugate $\overline{\omega}$ When we apply the inverse QFT, we get

$$\begin{aligned}
QFT^{-1}\left(\tfrac{1}{2^{t/2}} \textstyle\sum_{\ell=0}^{2^t-1} \omega^{\phi\ell}|\ell\rangle\right) &= \tfrac{1}{2^{t/2}} \textstyle\sum_{\ell=0}^{2^t-1} \omega^{\phi\ell} \sum_{k=0}^{2^t-1} \omega_t^{-\ell k}|\ell\rangle \qquad \text{(by (15))} \\
&= \tfrac{1}{2^{t/2}} \textstyle\sum_{\ell=0}^{2^t-1} \sum_{k=0}^{2^t-1} (\omega^{\phi-\ell 2^{-t}})^k|\ell\rangle.
\end{aligned}$$

The amplitude of $|\ell\rangle$ is therefore

$$\begin{aligned}
\alpha_\ell &:= \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} (\omega^{\phi-\ell 2^{-t}})^k \\
&= \frac{1}{2^{t/2}} \frac{1-(\omega^{\phi-\ell 2^{-t}})^{2^t}}{1-\omega^{\phi-\ell 2^{-t}}}
\end{aligned}$$

We use the fact that $2 \ge |1-e^{\mathbf{i}\theta}| \ge 2\theta/\pi$ for $|\theta| \le \pi$. Then $|1-\omega^x| = |1-e^{\mathbf{i}2pix}| \ge 4x$ when $|x| \le 1/2$. Hence

$$|\alpha_\ell| \le \frac{2}{2^t(1-\omega^{\phi-\ell 2^{-t}}}.$$

Writing β_ℓ for $\alpha_{2^t\widetilde{\phi}+\ell}$, we have

$$|\beta_\ell| \le \frac{2}{2^t(1-\omega^{\phi-\widetilde{\phi}-\ell 2^{-t}}} = \frac{2}{2^t(1-\omega^{\pm\delta-\ell 2^{-t}}} \le \frac{2}{2^t 4|\delta-\ell 2^{-t}|} = \frac{1}{2^{t+1}|\delta-\ell 2^{-t}|}.$$

Upon measurement of Register 1 at the end of the computation, suppose we obtain the state $|j\rangle = |j_1,\ldots,j_m\rangle$. The estimate for ϕ is therefore $j2^{-t}$. We want to upper bound the probability that $|j2^{-t}-\widetilde{\phi}| > 2^{-n}$ (this is the error probability because we want an n-bit approximation to ϕ). Writing $\Delta = 2^{t-n}$, we get

$$\begin{aligned}
\Pr\{|j-2^t\widetilde{\phi}| > 2^{t-n}\} &= \sum_{|\ell-2^t\widetilde{\phi}|\ge\Delta} |\alpha_\ell|^2 \\
&= \sum_{|\ell|\ge\Delta} |\beta_\ell|^2 \\
&\le \sum_{|\ell|\ge\Delta} \frac{1}{2^{t+1}|\delta-\ell 2^{-t}|^2} \\
&= \frac{1}{2}\sum_{|\ell|\ge\Delta} \frac{1}{|2^t\delta-\ell|^2} \\
&\le \sum_{\ell\ge\Delta} \frac{1}{(\ell-1)^2}, \qquad (2^t\delta \le 1) \\
&< \sum_{\ell\ge\Delta} \frac{1}{(\ell-1)(\ell-2)} \\
&= \sum_{\ell\ge\Delta} \left(\frac{1}{(\ell-1)} - \frac{1}{(\ell-2)}\right) \\
&< \frac{1}{\Delta-1}, \qquad \text{(by telescoping)}
\end{aligned}$$

We are almost there: recall that we want the probability of error to be at most $\varepsilon > 0$. Hence it is enough that $\varepsilon \ge 1/(\Delta-1) = 1/(2^{t-n}-1)$. Thus $t-n \ge \lg(1+\varepsilon^{-1})$. Since n and ε are user inputs, we only need to choose

$$t = n + \left\lceil \lg(1+\varepsilon^{-1}) \right\rceil. \tag{23}$$

We are done with phase estimation.

To summarize: given a blackbox U and an eigenvector $|v\rangle$, and a precision bound of n and error bound of $\varepsilon > 0$, if we construct the above 2-phase quantum circuit using the parameter t in Equation (23), then the measured value $0.j_1\cdots j_n$ in Register 1, is an n-bit approximation of ϕ with probability $1-\varepsilon$. Here, ϕ is given by $U|v\rangle = \omega^\phi|v\rangle$.

We can extend this technique to the case where an eigenvector $|v\rangle$ is not directly available, but we have a fixed state $|x\rangle = \sum_i c_i|v_i\rangle$ that is a superposition of eigenvectors $|v_i\rangle$, each with a phase ϕ_i. When we measure Register 1, for each i, we obtain an estimate of ϕ_i with probability $|c_i|^2$ (and on measuring Register 2, we find out the $|v_i\rangle$ whose phase we measured).

22.3.4 Integer Factoring Problem

We return to the main goal of integer factorization. The problem is easy to state: given an integer $N > 2$, to find distinct primes p_i and exponents $e_i \ge 1$ ($i = 1,\ldots,m$) such that

$$N = \prod_{i=1}^{m} p_i e^i, \tag{24}$$

as guaranteed by the Fundamental Theorem of Arithmetic (see below). This problem can easily be solved in time polynomial in N (see Exercise), but this is not considered a polynomial-time algorithm because the size of the input is only $n = \lfloor \lg(1+N) \rfloor$ (tbe number of bits in the binary representation of N). The current record for integer factorization takes time $\Theta((N\log^2 N)^{1/3})$ using sophisticated number field sieve methods. Our goal is to describe a quantum factoring algorithm that takes time $\Theta(\log^4 N) = \Theta(n^4)$.

Some Number Theory. When we say "numbers" in Number Theory, it means a natural number $n \in \mathbb{N}$. The starting point of Number Theory is the **divisibility relation** on integers: we say m **divides** n, and write $m|n$, if $ma = n$ for some $a \in \mathbb{Z}$. Also, let $m \nmid n$ denote the negation of divisibility. We say $n \in \mathbb{Z}$ is **prime** if it is divisible by exactly two numbers. This definition excludes the numbers 0 (being divisible by all numbers) and 1 (being divisible by one number). Hence 2 is the smallest prime number, and it also has the distinction of being the only even prime. The Fundamental Theorem of Arithmetic says that every number $n \geq 2$ has a unique representation of the form

$$n = \prod_{i=1}^{m} p_i^{e_i}, \qquad m \geq 1 \tag{25}$$

where $p_1 < p_2 < \cdots p_m$ are distinct primes, and $e_i \geq 1$. We write $x \equiv y (\text{mod}\, n)$ if $n|(x-y)$. We assume that students are familiar with the concept of **modulo n arithmetic**: we may add and multiply modulo n. E.g., $5+6 = 2(\text{mod}\, 9)$. Thus, the set $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ is a ring under modulo n arithmetic. We shall write $x \,\mathbf{mod}\, n$ for the unique value in $\mathbb{Z}_n$ that is equivalent to x modulo n. E.g., $11 \,\mathbf{mod}\, 9 = 2$.

The greatest common divisor (GCD) of m, n is denoted $\texttt{GCD}(m,n)$, and is the largest number that divides both m and n. When $m = n = 0$, we define $\texttt{GCD}(m,n) = 0$; otherwise, it is clear that $\texttt{GCD}(m,n) \geq 1$. If $\texttt{GCD}(m,n) = 1$, we say m, n are **coprime** (or, m is coprime to n). We can compute $\texttt{GCD}(m,n)$ in polynomial time, for instance, using Euclid's algorithm. Euclid's algorithm is simple to describe: if $n_0 > n_1 \geq 1$ are numbers, then we compute $\texttt{GCD}(n_0, n_1)$ be generating the following "Euclidean sequence"

$$(n_0, n_1, n_2, \ldots, n_{k-1}, n_k), \qquad k =\geq 1, \tag{26}$$

where

$$n_{i+1} = n_{i-1} \,\mathbf{mod}\, n_i, \qquad i = 1, \ldots, k-1. \tag{27}$$

The termination condition for the sequence is given by

$$n_{k-1} \,\mathbf{mod}\, n_k = 0. \tag{28}$$

It is easily seen from (27) that the following holds:

$$\texttt{GCD}(n_i, n_{i+1}) = \texttt{GCD}(n_{i-1}, n_i).$$

But $n_k = \texttt{GCD}(n_{k-1}, n_k)$, by the termination condition (28) Hence n_k is equal to $\texttt{GCD}(n_0, n_1)$. This proves the correctness of Euclid's algorithm. E.g., $\texttt{GCD}(22, 15) = 1$ follows from the Euclidean sequence

$$22, 15, 7, 1.$$

Again, $\texttt{GCD}(22, 18) = 2$ because $(22, 14, 8, 6, 2)$ is an Euclidean sequence. We can extract a valuable piece of information from the Euclidean algorithm: it is easy to verify by induction that in the Euclidean sequence (26), there exists integers s_i, t_i such that

$$n_{i+1} = s_i n_0 + t_i n_1, \qquad i = 1, \ldots, k-1.$$

In particular, there exists $s, t \in \mathbb{Z}$ such that $\texttt{GCD}(m,n) = sm + tn$. Thus $sm \equiv \texttt{GCD}(m,n)(\text{mod}\, n)$. When m, n are coprime, we conclude that $sm \equiv 1(\text{mod}\, n)$ and $tn \equiv 1(\text{mod}\, m)$. Thus s is the (multiplicative) **inverse** of m modulo n, and similarly t is the inverse of n modulo n. We sometimes write $s = m^{-1}(\text{mod}\, n)$. Summarizing: *every element m that is coprime to n has an inverse modulo n.* It is easy to modify the Euclidean algorithm to compute s_k, t_k as well (Exercise). This is usually called the Extended Euclidean algorithm. Thus we can compute multiplicative inverses.

For $n \in \mathbb{N}$, let

$$\mathbb{Z}_n^* = \{i \in \mathbb{Z}_n : \texttt{GCD}(i, n) = 1\}.$$

Since every element in $\mathbb{Z}_n^*$ has an inverse modulo n, and 1 is clearly the identity of multiplication modulo n, we conclude: *$\mathbb{Z}_n^*$ is a group under multiplication modulo n.* The size of this group is $|\mathbb{Z}_n^*| = \phi(n)$ where $\phi(n)$ is Euler's totient function. Thus $\phi(n)$ is the number of distinct values in $\mathbb{Z}_n$ that are coprime to n. For instance, if p is prime, $\phi(p) = p-1$ since $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$. We can easily generalize this to $\phi(p^e) = p^{e-1}(p-1)$ because the elements in $\mathbb{Z}_{p^e} \setminus \mathbb{Z}_{p^e}^*$ are precisely the multiples of p and there are p^{e-1} of these. Thus $\phi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1)$. Further, if m, n are coprime, we can show (Exercise) that $\{xx' : x \in \mathbb{Z}_n^*, x' \in \mathbb{Z}_m^*\} = \mathbb{Z}_{mn}^*$. Hence $\phi(mn) = \phi(m)\phi(n)$. This gives us a formula for $\phi(n)$ when we know the complete factorization of n, similar to (25). For instance, $\phi(24) = \phi(2^3)\phi(3) = 8$. The Fermat-Euler theorem says that for all $x \in \mathbb{Z}_n^*$, $x^{\phi(n)} = 1(\text{mod}\, n)$. A **power** is a number of the form m^e for some $m, e \in \mathbb{N}$. When m is prime, then m^e is called a **prime power**.

Reductions of the Factoring Problem. Henceforth, we simply say "factorization" for "integer factorization". We reduce the factorization problem to its computational "core". Here, we say a problem P is reducible to another problem Q if we can construct a polynomial time algorithm A_P for P from any polynomial time algorithm A_Q for Q. Moreover, this reduction is randomized if A_P is randomized (regardless of whether A_Q is randomized or not). In general we need randomized algorithms.

1. **Reduction to Simple Factoring:** First we may reduce the problem to finding any non-trivial factor M of N. That is, find M such that $M|N$ and $1 < M < N$. Call this the **simple factoring problem**, as opposed to the original version which we call the **complete factoring problem**, represented by (25). The complete factoring problem is reduced to at most $\lg N$ simple factoring problem. This is because in (25), we have $N = \prod_{i=1}^m p_i^{e_i} \ge \prod_{i=1}^m 2^{e_i} \ge 2^e$ where $e = \sum_{i=1}^m e_i$. Hence $e \le \lg N$.

2. **Reduction to an Odd Non-power:** Given an N, we want to detect if it is a power, and if so, completely factor it, $N = M^e$. This can easily be done in polynomial time (Exercise).

   We can further assume N is odd. This is trivial, but the reader will rightfully wonder if this is a just an adhoc decision: why only exclude multiples of 2? We could likewise exclude any multiples of 3 or 5, or any number we like. The reason this decision is not adhoc is related to the next reduction.

3. **Reduction to Finding a Squareroot of Unity modulo N.** Call x a **squareroot** of unity (*i.e.*, 1) modulo N if $x^2 \equiv 1 (\bmod N)$. Clearly, $x = 1$ and $x = N - 1$ (usually denoted -1) are squareroots of 1 modulo N. But these are the **trivial squareroots** of 1; we want nontrivial squareroots where $1 < x < N - 1$. Armed with such an x, we see that $x^2 - 1 = (x - 1)(x + 1)$ must be divisible by N. Hence, a prime factor p of N must divide either $x - 1$ or $x + 1$. This means p divides either $\texttt{GCD}(N, x - 1)$ or $\texttt{GCD}(N, x + 1)$, *i.e.*, either $\texttt{GCD}(N, x - 1)$ or $\texttt{GCD}(N, x + 1)$ is a nontrivial factor of N.

4. **Reduction to Order Finding.** We can reduce finding non-trivial squareroots of unity to order finding. For $x \in \mathbb{Z}_n^*$, the **order of x modulo n** (or, the **n-order of** x, or $\texttt{ord}_n(x)$) is the smallest $r \ge 0$ such that $x^r = 1 (\bmod n)$. The **order finding problem** is to find r given x, n. By the Euler Fermat theorem, $x^{\phi(n)} \equiv 1 (\bmod n)$ and hence $r \le \phi(n)$. Indeed, $r|\phi(n)$ because if $\phi(n) = ar + b$ where $0 \le b < r$, then $x^b \equiv x^{b+ar} = x^{\phi(n)} \equiv 1 (\bmod n)$, which is a contradition unless $b = 0$. The order finding problem is not known to be solvable in polynomial time. The reduction to order finding is nontrivial and will be taken up below.

Squareroots of Unity. The above reduction prompts a closer examination of the squareroots of 1. Let x be a squareroot of 1 modulo n. An obvious question is: when is x nontrivial? To answer this question, we must look into the group structure of $\mathbb{Z}_n^*$.
1. The simplest kind of groups[6] are the cyclic groups. By definition, a group Z is cyclic if there exists $g \in Z$ (called a **generator**) that generate the entire group by repeated multiplication: $Z = \{g^i : i \in \mathbb{N}\}$. It is known that $\mathbb{Z}_n^*$ is cyclic if and only if $n = 2, 4$ or $n = p^m$ or $2p^m$ where p is an odd prime and $m \ge 1$.
2. *If $\mathbb{Z}_n^*$ is cyclic then the only squareroots of 1 modulo n are the trivial ones.* In proof, let $g \in \mathbb{Z}_n^*$ be a generator of the cyclic group and let $x = g^e$ for some e $(0 < e < \phi(n))$. Since $g^{2e} \equiv x^2 \equiv 1 (\bmod n)$, we have $\phi(n)|2e$ and so $\phi(n)/2 \le e$. But $\phi(n)/2 < e$ is not possible because it would lead to the contradiction that $x^{2e-\phi(n)} \equiv 1$ and $1 \le 2e - \phi(n) < \phi(n)$. This proves our claim.
4. *The number of generators in a cyclic group Z of size n is $\phi(n)$.* To see this, let g be any generator of Z. We claim that g^e is also a generator whenever $\texttt{GCD}(e, n) = 1$. For, in this case, if r is the order of g^e then $g^{re} = 1$ implies $n|re$ and hence $n|r$. We conclude that $r = n$. This concludes our proof because there are $\phi(n)$ choices for e. As corollary, $\mathbb{Z}_{q_i}^*$ is $\phi(\phi(q_i)) = \phi(p_i^{e_i-1}(p_i - 1))$.

Example. Suppose $n = 45 = 9 \times 5$. Then $\phi(n) = \phi(9)\phi(5) = 6 \times 4 = 24$. How, $\phi^2(9) = \phi^2(5) = 2$ and so $\mathbb{Z}_9^*$ and $\mathbb{Z}_9^*$ each has 2 generators each. We may check that 2 and 5 are the generators of $\mathbb{Z}_9^* = \{1, 2, 4, 5, 7, 8\}$, and 2 and 3 are generators of $\mathbb{Z}_5^* = \{1, 2, 3, 4\}$.

Group Structure of Multiplicative Group of n. We may now assume that n is an odd non-power. Then $n = \prod_{i=1}^m q_i$ $(m \ge 2)$ where each q_i is a prime power and the q_i's are coprime.
1. The group $\mathbb{Z}_n^*$ is thus isomorphic to

$$G = \mathbb{Z}_{q_1}^* \times \cdots \times \mathbb{Z}_{q_m}^*$$

because of the Chinese Remainder Theorem. Indeed the isomorphism is the natural one, $h : \mathbb{Z}_n^* \to G$ where $h(x) = (x_1, \ldots, x_n)$ and $x_i \equiv x (\bmod q_i)$. Let $\overline{h}$ denote the inverse of h.

[6] We assume only the basic definition of groups.

2. The group operation $\circ$ in G is componentwise multiplication, modulo q_i in the ith component. Thus, $h(xy) = h(x) \circ h(y)$. Let us note that

$$h(1) = (1, 1, \ldots, 1), \qquad h(-1) = (-1, -1, \ldots, -1).$$

Let $h(x) = (x_1, \ldots, x_m)$. Then for any $r \geq 0$, $h(x^r) = (x_1^r, \ldots, x_m^r)$, which we write simply as $(x_1, \ldots, x_m)^r$.
3. Let g_i be a generator of $\mathbb{Z}_{q_i}^*$, and $x_i = g_i^{e_i}$ for some $e_i \geq 0$. CLAIM: *If the n-order of x and x_i (respectively) are r and r_i, then r is equal to $\ell := \mathtt{LCM}(r_1, \ldots, r_m)$.* To see this, first note that $\phi(n)|r$ and $\phi(q_i)|r_i$. Also, $x_i^r \equiv 1 (\bmod\, n)$ implies $x_i^r \equiv 1(\bmod\, q_i)$ and so $r_i|r$. Thus $r \geq \ell$. But $x^\ell \equiv 1(\bmod\, n)$ implies $r \leq \ell$. This proves our claim that $r = \ell$.
4. *The fraction of elements in $\mathbb{Z}_n^*$ of odd order modulo n is most 2^{-m}.* From $x^r \equiv 1(\bmod\, n)$, we conclude that $g_i^{e_i r} \equiv 1(\bmod\, q_i)$ and so $\phi(q_i)|e_i r$. If e_i is odd, then r must be even. Therefore, a necessary condition for r to be odd is that every e_i be even. The fraction of elements x in $\mathbb{Z}_n^*$ such that $x \mapsto (e_1, \ldots, e_m)$ with all e_i even is exactly $1/2^m$.

Randomized Order Finding. We introduce a useful notation: for $n \in \mathbb{N}$, let $\mathrm{v}(n) = \mathrm{v}_2(n)$ be the largest $d \geq 0$ such that $2^d|n$. This function is not defined for $n = 0$. We can generalize this to any nonzero rational number n/m: $\mathrm{v}(n/m) := \mathrm{v}(n) - \mathrm{v}(m)$, which can be negative or positive.
1. Now let $\mathbb{Z}_n^k := \{i \in \mathbb{Z}_n : i > 0, \mathrm{v}(i) = k\}$ and $f_n(k) = |\mathbb{Z}_n^k|/n$. For instance, $f_{16}(k) = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, 0$ for $k = 0, 1, 2, 3, 4$. We may easily verify, for all $k \geq 0$:

$$\begin{aligned}
f_n(0) &\leq 1/2, \text{ with equality iff } n \text{ even,} \\
\sum_{\ell \geq 1} f_n(\ell) &\leq 1/2, \text{ with equality iff } n \text{ odd.} \\
f_n(k+1) &\leq f_n(k) \\
f_n(k) &\leq \frac{1}{2^k + 1} \\
f_n(k) &\geq \frac{1}{3 \cdot 2^k}, \text{ provided } f_n(k) > 0.
\end{aligned}$$

2. In our application, we are interested in the following fraction: let $m, n, k \in \mathbb{N}$. Then define $\mathbb{Z}_{m,n}^k := \{(d, e) \in \mathbb{Z}_m \times \mathbb{Z}_n : d > 0, e > 0, \mathrm{v}(d) - \mathrm{v}(e) = k\}$, and $f_{m,n}(k) = |\mathbb{Z}_{m,n}^k|/mn$. Clearly,

$$\begin{aligned}
f_{m,n}(k) &= \sum_{\ell \geq 0} f_m(k+\ell) f_n(\ell) \qquad (29) \\
f_{m,n}(k+1) &\leq f_{m,n}(k) \\
f_{m,n}(k) &\leq \frac{947}{1800} < 5/9
\end{aligned}$$

We prove the last inequality: It is enough to show that $f_{m,n}(0) \leq \frac{947}{1800}$. From above, we know that $f_m(0) \leq 1/2$, $f_m(1) \leq 1/3$, $f_m(2) \leq 1/5$ and $f_m(k) \leq 2^{-k}$. Thus,

$$\begin{aligned}
f_{m,n}(0) &= \sum_{\ell \geq 0} f_m(k+\ell) f_n(\ell) \\
&= f_m(0)f_n(0) + f_m(1)f_n(1) + f_m(2)f_n(2) + \sum_{\ell \geq 3} 2^{-\ell-1} \\
&= \frac{1}{4} + \frac{1}{9} + \frac{1}{25} + \frac{1}{8} \\
&= \frac{947}{1800}.
\end{aligned}$$

3. Let $E = \{x \in \mathbb{Z}_n^* : \mathtt{ord}_n(x) = \text{even}\}$. From the preceding, we know that $|E| \geq \phi(n)(1 - 2^{-m})$. If $x \in E$ and $\mathtt{ord}(x) = r$, let $s(x) = x^{r/2} \,\mathbf{mod}\, n$. Thus $s(x)$ is a squareroot of unity. CLAIM: *The fraction of elements in E such that $s(x)$ is a nontrivial squareroot of unity is at least $4/9$.* We know that $s(x) \neq 1$. Therefore it suffices to show that for less than $5/9$ of the elements x in E, $s(x) = -1$. If r is the order of x, and writing $h(x) = (g_1^{e_1}, \ldots, g_m^{e_m})$ as before, showing $s(x) = -1$ amounts to showing

$$g_i^{e_i r/2} \equiv -1(\bmod\, q_i)$$

for all i. Now $g_i^{e_i r} \equiv 1 (\bmod q_i)$ implies $\phi(q_i) | e_i r$, and so

$$\mathrm{v}_2(\phi(q_i)) \le \mathrm{v}_2(e_i r). \tag{30}$$

If $g_i^{e_i r/2} \equiv -1 (\bmod q_i)$ then $\phi(q_i) \nmid e_i r/2$, and so

$$\mathrm{v}_2(\phi(q_i)) > \mathrm{v}_2(e_i r/2). \tag{31}$$

From (30) and (31), we conclude $e_i \ge 1$ and $\mathrm{v}_2(\phi(q_i)) = \mathrm{v}_2(e_i r)$ for all i. Since $m \ge 2$,

$$\mathrm{v}_2(e_1/e_2) = \mathrm{v}_2(\phi(q_1)/\phi(q_2)).$$

The righthand side is a constant k. Without loss of generality, assume $k \ge 0$. The fraction of $(e_1, e_2) \in \{1, \ldots, \phi(q_1) - 1\} \times \{1, \ldots, \phi(q_2) - 1\}$ such that $\mathrm{v}_2(e_1/e_2) = \mathrm{v}_2(e_1) - \mathrm{v}_2(e_2) = k$ is less than 5/9, according to (30).

4. *With probability at least* $1/3$, *a random element* $x \in \mathbb{Z}_n^*$ *have even order, with* $s(x)$ *a nontrivial squareroot of unity.* Let A be the event that x has even order. Let B be the event that $s(x)$ is a nontrivial squareroot of unity. The claimed probability is equal to $\Pr(AB) = \Pr(B|A)\Pr(A)$. But above, we have shown that $\Pr(A) \ge 1 - 2^{-m} \ge 3/4$, and $\Pr(B|A) \ge 4/9$. Multiplying these probabilities gives our claim: $(3/4)(4/9) = 1/3$.

5. We have thus reduced the simple factorization problem to order finding: given N to be factored, we may assume N is a non-power odd number. We randomly choose $x \in \mathbb{Z}_N$. We must check if $x \in \mathbb{Z}_N^*$, by computing $\mathtt{GCD}(x, N)$. If this GCD is not 1, we have in fact found a factor! Hence we may assume $x \in \mathbb{Z}_N^*$ and proceed to find its N-order r. If N is composite, with probability more than $1/3$, r would be even and $s(x) = x^{r/2}$ a non-trivial squareroot of 1 modulo N. Thus with probability more than $1/3$ we can factor N. In the contrary case, we can repeat this test $k \ge 2$ times. If we fail to factor N for k times, we declare N to be prime. What is the probability of error? Error can only occur if N is composite and we declare it prime. But this happens only if we fail the test for k times. This probability is at most $(2/3)^k$, which can be as small as we like by making k large enough. For instance $k = 4$, will ensure that failure probability of less than 20%. This proves:

THEOREM 5 *If there is a randomized polynomial time algorithm for order finding, then there is a randomized polynomial time algorithm for integer factorization.*

Example (contd). Continue with $n = 45 = 9 \times 5$. Let $r = \mathtt{LCM}(\phi(9), \phi(5)) = 12$. Consider x such that $h(x) = (2, 2)$. We have $2^6 \equiv 5^6 \equiv 1 (\bmod 9)$ and also $2^6 \equiv 3^6 \equiv -1 (\bmod 5)$. Hence $h(x^6) = (1, -1)$. But what is x^6? Well, $x^6 \equiv 1 (\bmod 9)$ means $(x^6 \,\mathbf{mod}\, 45) \in \{1, 10, 19, 28, 37\}$. Similarly, $x^6 \equiv -1 (\bmod 5)$ implies $(x^6 \,\mathbf{mod}\, 45) \in \{4, 9, 14, 19, 24, 29, 34, 39, 44\}$. This means $x^6 = 19$. Check: Modulo 45, we have $19^2 = (20 - 1)^2 = 400 - 40 + 1 \equiv -50 + 5 + 1 \equiv 1$. Thus $y = 19$ is a nontrivial square root of unity. Our reduction tells us that $\mathtt{GCD}(45, y - 1)$ or $\mathtt{GCD}(45, y + 1)$ must be nontrivial. Indeed, $\mathtt{GCD}(45, y - 1) = 9$ and $\mathtt{GCD}(45, 20) = 5$.

The preceding development shows why assuming n is odd in our reduction of the factorization problem is not an arbitrary decision: it ensures that each $\mathbb{Z}_{q_i}^*$ is cyclic.

22.3.5 Quantum Order Finding

Let $N = 2^n$ be fixed. Given $3 \le m < N$ and $x \in \mathbb{Z}_m^*$, we want to find the m-order of x. For instance, $N = 16 = 2^4$, $m = 15$ and $x = 7$. Then the 15-order of x is $r = 4$. But how can we produce r from m, x using a quantum algorithm? The trick is to define a unitary operator $U_{m,x}$ and an eigenvector $|v\rangle$ such that $U_{m,x}(|v\rangle) = \omega^{f(r)}|v\rangle$ where $f(r)$ is some easily inverted function of r. Then by phase estimation, we can approximate $f(r)$ and then invert $f(r)$ to get r.

1. The unitary operator $U_{m,x}$ will act on the usual state space of n qubits, with the pure states $|0\rangle, \ldots, |N-1\rangle$. This operator is completely described by its actions on the pure states

$$U_{m,x}(|j\rangle) := \begin{cases} |jx \,\mathbf{mod}\, m\rangle & \text{if } j \in \mathbb{Z}_m \\ |j\rangle & \text{else.} \end{cases} \tag{32}$$

Since x has an inverse modulo m, $jx \equiv j'x (\bmod m)$ implies $j \equiv jxx^{-1} \equiv j'xx^{-1} \equiv j' (\bmod m)$. Thus $U_{m,x}$ is a permutation matrix, and *á fortiori*, a unitary matrix.

2. We next find eigenvectors of $U_{m,x}$. For each $s \in \mathbb{Z}_r$, let

$$|v_s\rangle := \frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} \omega^{-s\ell/r} |x^\ell \,\mathbf{mod}\, m\rangle. \tag{33}$$

Then

$$\begin{aligned} U_{m,x}(|v_s\rangle) &= \frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} \omega^{-s\ell/r} |x^{1+\ell} \,\mathbf{mod}\, m\rangle \\ &= \frac{1}{\sqrt{r}} \omega^{s/r} \sum_{\ell=0}^{r-1} \omega^{-s(1+\ell)/r} |x^{1+\ell} \,\mathbf{mod}\, m\rangle \\ &= \omega^{s/r} |v_s\rangle. \end{aligned}$$

Thus, $|v_s\rangle$ is an eigenvector of $U_{m,x}$ with phase $\phi = s/r$. If we can estimate ϕ, and assuming we know s, we can trivially recover r, provided $s \neq 0$. Unfortunately, we do not know how to prepare the state $|v_s\rangle$ for any s. To circumvent this problem, we use another observation: for any k,

$$\begin{aligned} \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \omega^{sk/r} |v_s\rangle &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \omega^{sk/r} \frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} \omega^{-s\ell/r} |x^{\ell} \,\mathbf{mod}\, m\rangle \\ &= \frac{1}{r} \sum_{\ell=0}^{r-1} \sum_{s=0}^{r-1} \omega^{(k-\ell)s/r} |x^{\ell} \,\mathbf{mod}\, m\rangle \\ &= |x^k \,\mathbf{mod}\, m\rangle \end{aligned}$$

where the last equation follows from the fact that (see (12)) $\sum_{s=0}^{r-1} \omega^{(k-\ell)s/r}$ vanishes for $\ell \neq k$, and otherwise equals r.
3. To apply the previous result, we need the state $|x^k \,\mathbf{mod}\, m\rangle$ for each k. Hence, we want to construct the exponentiation transformation,

$$|k\rangle|y\rangle \mapsto |k\rangle|x^k y \,\mathbf{mod}\, m\rangle.$$

The circuit is shown in Figure 22.10, where U refers to $U_{m,x}$.

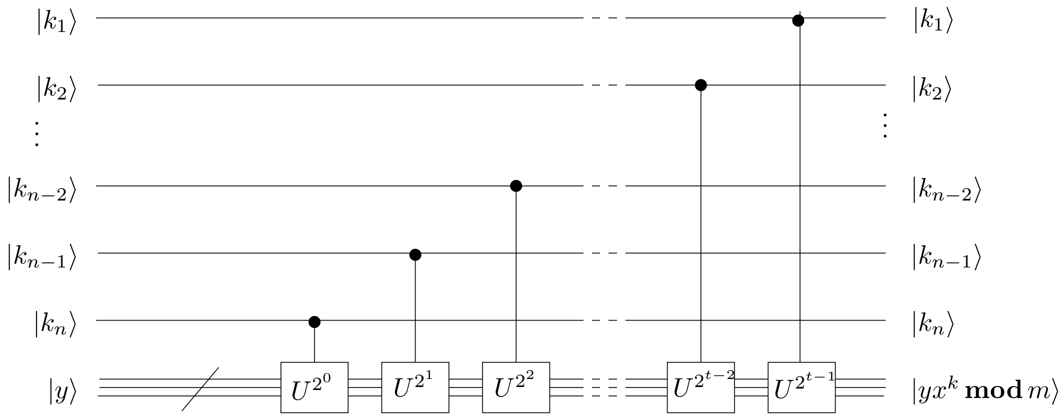


Figure 22.10: Exponentiation Circuit for $|k\rangle]|y\rangle \mapsto |k\rangle|x^k y \,\mathbf{mod}\, m\rangle$.

Note that this circuit is similar to the first stage of phase estimation. What we must remember, however, is that the U^{2^i}-gates must be implemented efficiently (polynomial in i, not in 2^i). This uses the well-known successive squaring trick of classical exponentiation. We leave this as an Exercise.
4. Putting it Together. Let us start with two quwords, each with n bits. Prepare them as follows:

$$|0\rangle \otimes |1\rangle.$$

Applying the Hadarmard transformation H on the first quword, we obtain

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle \otimes |1\rangle.$$

Next, apply the exponentiation operator above the two q-words to obtain

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}|k\rangle\otimes|x^k\,\mathbf{mod}\,m\rangle.$$

But this last result can also be expressed as

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\left(|k\rangle\otimes\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\omega^{sk/r}|v_s\rangle\right) \quad = \quad \frac{1}{\sqrt{rN}}\sum_{s=0}^{r-1}\left(\sum_{k=0}^{N-1}|k\rangle\otimes\omega^{sk/r}|v_s\rangle\right).$$

The expression in the final pair of parentheses is similar to a Fourier transform, but with phase $\omega^{s/r}$. Indeed, we can verify that by applying an inverse Fourier transform (in analogy to the second stage of phase estimation), we will obtain estimates $\widetilde{\phi}_s=\widetilde{\phi}$ for $\phi_2=s/r$. Note that we obtain estimates $\widetilde{\phi}_s$ for each choice of s. When $s=0$ or $\mathtt{GCD}(s,r)>1$, the answer will not be correct, but otherwise, we are indifferent as to which s we obtain. Since at least $Cr/\ln(r)$ numbers less r are prime, the probability of obtaining an s that is coprime to r is at least $C/\ln(r)$. So if we repeat at least $2C/\ln(r)$ times, we have a strong change of obtaining a good s.
5. Final Touch. Once we have an estimate of s/r to sufficient accuracy, we use a well-known fact about rational approximation: if $|\widetilde{\phi}-s/r|\le 1/(2r^2)$, then we can recover s/r uniquely, using the simple continued fraction algorithm. Of course, we do not know r, but only need at upper bound on r (e.g., N will do). This part of the computation does not need any quantum power.

Exercise

Exercise 22.3.1: Let $0<\varepsilon<1$. Give an randomized classical algorithm to solve the Deutsch-Jozsa problem with probability of success of $1-\varepsilon$ using $O(\log(1/\varepsilon))$ calls to the blackbox. HINT: Let E_i $(i=0,1)$ be the event that $f(x_1,\ldots,x_n,1)=i$ where $x_1,\ldots,x_n$ are random. What is the probability that E_0 occurs in k trials? □

Exercise 22.3.2: Give a simple upper bound on the number of classical gates to compute DFT (assuming each gate can perform a single complexity arithmetic operation in constant time). □

Exercise 22.3.3: We want to estimate the phase ϕ_1 of an eigenvector $|v_1\rangle$ of a blackbox unitary operator U. We outlined a method to estimate ϕ_1 when we can only prepare a state $|x\rangle$ that contains $|v_1\rangle$ as one of its components. Compute the probability of correct measurements of ϕ_1, and discuss its impact on complexity. □

Exercise 22.3.4: Describe an algorithm for the complete factorization of an integer $N\in\mathbb{N}$ that runs in time polynomial in N. □

Exercise 22.3.5: Describe a polynomial time algorithm which, given $N\in\mathbb{N}$, either detects that N is not a power or else completely factorize N, *i.e.*, finds $M,e\in\mathbb{N}$ such that $N=M^e$. HINT: how would you detect if N is a square, $N=M^2$? □

Exercise 22.3.6: (Extended Euclidean Algorithm) Modify the Euclidean algorithm to compute s,t,d for any input numbers m,n, such that $d=\mathtt{GCD}(m,n)$ and $d=sm+tn$. □

Exercise 22.3.7: If n is an odd non-power, then $\mathbb{Z}_n^*$ is non-cyclic. □

Exercise 22.3.8: Let $x\in\mathbb{Z}_n^*$ and $n=q_1\cdots q_m$ $(m\ge 2)$ where the q_i's are prime powers, and coprime to each other. Assume $h(x)=(g_1,\ldots,g_m)$ where each g_i is a generator of $\mathbb{Z}_{q_i}^*$. Characterize the conditions where $x^\ell\equiv -1(\mathrm{mod}\,n)$ where $2\ell=\mathtt{LCM}(\phi(q_1),\ldots,\phi(q_m))$. □

Exercise 22.3.9: Recently, it was announced that a quantum computer was able to factor the number 15. Deduce what probably happened – what was, and what was not done by the quantum computer. □

Exercise 22.3.10: Let q be a prime power and $d=\mathrm{v}_2(\phi(q))$. Then exactly half of the elements in $\mathbb{Z}_q^*$ has q-order that is divisible by 2^d. □

End Exercise

22.4 Quantum Turing Machines

A circuit computes a finite function. We now address general computational models that takes inputs of arbitrarily large size. In complexity theory, we can take one of two paths. One way is to define a general computing model such as Turing machines. Another way is to start from circuits, and to define[7] a "uniform circuit family". These two approaches can also be taken to define a more general computational model for quantum computing. A basic result here is the existence of a universal Turing machine. Deutsch [14] proved the analogous result for quantum computers. Bernstein and Vazirani [11] describes a similar model, usually known as the quantum Turing machine (QTM). Benioff [6] has argued for a different basis for such models.

Need to give the results of Yao??

22.5 Quantum Choices

We can now generalize our choice model to incorporate quantum complexity as follows. Let δ be a transition table for a Turing machine as usual. We need to define functions γ for each state q in δ and provide an acceptance rule.

Some of the justification of our procedure has been pointed out by Lance Fortnow in his paper "A Theoreticians View of Quantum Computing".

For each state q, $\gamma(q)$ will be a unitary ???

APPENDIX: Review of Linear Algebra

We review of some facts from linear algebra needed in quantum computing. We work exclusively with finite-dimensional linear spaces over the complex field $\mathbb{C}$. The (complex) conjugate $\overline{z}$ of a complex number $z \in \mathbb{C}$ is denoted $\overline{z} = x - \mathbf{i}y$ where $z = x + \mathbf{i}y$ with $x, y \in \mathbb{R}$ and $\mathbf{i} = \sqrt{-1}$. We will shortly see that the complex conjugate of z can also be denoted z^*, and this form is common in quantum physics. The **absolute value** $|z|$ of z is equal to $\sqrt{z\overline{z}} = \sqrt{x^2 + y^2}$. Complex numbers with absolute value 1 are called **complex signs**, a generalization of the real signs ± 1. If $|z| \le 1$, we call z a (probability) **amplitude**. A complex sign z can be written in the form $e^{\mathbf{i}\theta}$ for some real θ. The value θ is also called the **phase**.

A matrix $A \in \mathbb{C}^{m\times n}$ is viewed as a transformation $t_A : \mathbb{C}^n \to \mathbb{C}^m$, where $t_A(x) = Ax$ (a matrix-vector multiplication). Thus, $range(A)$ is simply $\{Ax : x \in \mathbb{C}^n\} \subseteq \mathbb{C}^m$. Let us assume $m = n$ unless otherwise noted. The (i, j)-th entry of a matrix A is denoted $(A)_{ij}$. The **transpose** A^T and **conjugate** $\overline{A}$ of a matrix A is given by $(A^T)_{ij} = (A)_{ji}$ and $(\overline{A})_{ij} = \overline{(A)_{ij}}$, respectively. Then the **conjugate transpose** A^* is given by $A^* = \overline{A^T} = \overline{A}^T$. Note that $(AB)^T = B^T A^T$ and $(AB)^* = B^* A^*$. A matrix $A \in \mathbb{C}^{n\times n}$ is **Hermitian** if $A^* = A$, **unitary** if $A^*A = I$ (identity), and **orthogonal** if $A^T A = I$. So if A is unitary then $A^{-1} = A^*$, $A^*A = AA^*$. Hermitian matrices are also known as "self-adjoint" matrices (as A^* is sometimes called the "adjoint" of A). In case A is a 1×1 matrix, A^* is just another way of writing complex conjugation, since $A^* = \overline{A}$. Unitary matrices are fundamental in quantum computing. For a unitary U, it is clear that $\det(U) = 1$ and hence its eigenvalues λ_i are complex signs, $|\lambda_i| = 1$.

A matrix A is **normal** if $A^*A = AA^*$. Note that unitary matrices, Hermitian matrices, skew-Hermitian matrices ($A^* = -A$) are all normal. In quantum mechanics, Hermitian and unitary matrices are of paramount importance.

Orthogonalization and QR-Factorization. A useful tool for investigating the structure of linear spaces is based on a certain factorization of matrices. Let $A = [a_1|a_2|\cdots|a_m] \in \mathbb{C}^{n\times m}$ where a_i is the ith column. Let $S_i \subseteq \mathbb{C}^n$ be the subspace spanned by the first i columns.

Assume A has rank m (so $m \le n$). Let the sequence $(q_1, \dots, q_m)$ of vectors form an orthonormal basis for $S_m \subseteq \mathbb{C}^n$. If we form the matrix

$$Q := [q_1|q_2|\cdots|q_m],$$

then there is some $m \times m$ matrix R such that

$$A = QR. \tag{34}$$

The ith column r_i of R represents the vector a_i relative to the basis $(q_1, \dots, q_m)$. Let us call $(q_1, \dots, q_m)$ (or Q) a **Gram-Schmidt basis** for A if for each $i = 1, \dots, m$, the prefix $(q_1, \dots, q_i)$ forms an ordered basis for S_i. In this case, the matrix R is upper triangular in (34). The well-known Gram-Schmid orthogonalization procedure that compute such a basis Q from any A.

[7] There is a bit of circularity in the conventional definition of "uniformity" via some Turing machine (say). We can avoid this problem by using some logical circuit description language, say.

The factorization (34) of A is known as a **reduced QR-factorization** when Q is an Gram-Schmidt basis of A. Each vector q_i in this basis is unique up to some scalar multiple of modulus 1. Equivalently, we say q_i is determined up to "complex signs". To make the factorization unique, we choose the complex signs to make the diagonal elements of R real and non-negative. The **full QR-factorization** of A is the following variant,

$$A = \widehat{Q}\widehat{R} \tag{35}$$

where $\widehat{Q}$ is $n \times n$ and $\widehat{R}$ is $n \times m$. It is obtained from (34) by augmenting Q and R: the matrix $\widehat{Q}$ is obtained by appending $n - m$ additional columns so that the columns of $\widehat{Q}$ form an orthonormal basis for $\mathbb{C}^n$; the matrix $\widehat{R}$ is obtained by appending $n - m$ additional rows of 0's. Note that when $m = n$, the full QR-factorization is just the reduced QR-factorization. Since $\widehat{Q}^*\widehat{Q} = I$, the matrix Q is unitary in the full QR-factorization.

So far, we have assumed that A has rank m. Suppose m, n are arbitrary and A has has rank k (so $k \le \min\{m, n\}$. We first apply a permutation P to the columns of A so that the first k columns are linearly independent. Then we have $AP = QR$ where $Q \in \mathbb{C}^{n\times n}$ is unitary, $R \in \mathbb{C}^{n\times m}$ is upper triangular, and the first k columns of Q forms a orthonormal basis for S_m,

Eigenvalues and Eigenvectors. A non-zero vector $x \in \mathbb{C}^n$ is called an **eigenvector** of A if $Ax = \lambda x$ for some $\lambda \in \mathbb{C}$. In this case, we call λ an **eigenvalue** of A that is associated with the eigenvector x. Note that while eigenvectors must be non-zero, we have no such restriction on eigenvalues. In particular, A is singular iff $\lambda = 0$ is an eigenvalue: $Ax = 0x = 0$ ($x \ne 0$) iff A is singular. The set of eigenvalues of A, denoted $\Lambda(A)$, is called the **spectrum** of A. The **characteristic polynomial** of A is $p_A(z) = \det(zI - A)$.

For example, if $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then $p_A(z) = \det \begin{bmatrix} z-a & b \\ c & z-d \end{bmatrix} = (z-a)(z-d) - bc = z^2 - (a+d)z + ad - bc$. If

$$A = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & & \\ & & & 0 & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix}$$

then $p_A(z) = z^n + \sum_{i=0}^{n-1} a_i z^i$.

In general, it is easily seen that $p(z)$ is monic of degree n with the constant term equal to $\det(A)$ and the coefficient of z^{n-1} equal to $-trace(A) = -\sum_{i=1}^n a_{ii}$. Also, $\lambda \in \Lambda(A)$ iff λ is a zero of $p_A(z)$. [In proof, $Ax = \lambda x$ iff $(\lambda I - A)x = 0$ iff $\lambda I - A$ is singular iff $\det(\lambda I - A) = 0$.] The multiplicity of λ as a root of $p_A(z)$ is called the **algebraic multiplicity** of λ. It follows that the cardinality of $\Lambda(A)$ is between 1 and n. If A can be written as a block matrix of the form

$$A = \begin{bmatrix} B & C \\ \mathbf{0} & D \end{bmatrix}$$

where B and D are square blocks, then

$$p_A(z) = p_B(z)p_D(z). \tag{36}$$

This is equivalent to $\det(zI - A) = \det(zI - B)\det(zI - D)$. In proof, we only have to show that any of the $n!$ terms of the determinant $\det(zI - A)$ that involves an entry of C is zero. Suppose B is $k \times k$ and the term $t = \prod_{i=1}^n a_{i,j(i)}$ contains an entry $a_{i_0,j(i_0)}$ of C. Then the i_0th row of B does not contribute to t. If B contributes ℓ entries to t, this implies $\ell \le k - 1$. So for some $1 \le c \le k$, the c-th column of B does not contribute to t. Consider the index i_1 where $j(i_1) = c$: clearly, the factor $a_{i_1,j(i_1)}$ of t is 0 and so $t = 0$, concluding our proof. As corollary, we have $\Lambda(A) = \Lambda(B) \cup \Lambda(D)$.

Invariant subspaces. A subspace $E \subseteq \mathbb{C}^n$ is A**-invariant** if $AE = \{Ax : x \in E\}$ is contained in E. If $E = \{0\}$ then it is clearly A-invariant. We call this the trivial case. If $\lambda \in \Lambda(A)$, the set $E_\lambda = \{x \in \mathbb{C}^n : Ax = \lambda x\}$ is easily seen to be a non-trivial A-invariant subspace; we call E_λ the **eigenspace** of A associated with λ. In particular, the eigenspace of A associated with $\lambda = 0$ is the nullspace of A. If $\lambda \ne \lambda'$ then clearly $E_\lambda \cap E_{\lambda'} = \{0\}$. Further, $x \in E_\lambda$ and $y \in E_{\lambda'}$ are linearly dependent: for, if $c = ax + by = 0$ for some $a, b \in \mathbb{C}$ then $Ac = \lambda ax + \lambda' by = 0$, which easily implies $a = b = 0$. In E_λ, transformation by A amounts to scaling by a factor of λ. The dimension of E_λ is called the **geometric multiplicity** of λ. We will show below that the geometric multiplicity of λ is at most the algebraic multiplicity.

Similarity. Invariant subspaces are intimately connected to the notion of "similarity". Two matrices $A, B \in \mathbb{C}^{n\times n}$ are **similar** if $A = XBX^{-1}$ for some non-singular matrix X. In case X is unitary, $X^*X = I$, we say A and B are **unitarily similar**: $A = XBX^*$. Similar matrices A, B have the same characteristic polynomial since

$$\det(zI - X^{-1}BX) = \det(X^{-1}(zI - B)X) = \det(X^{-1})\det(zI - B)\det(X) = \det(zI - B).$$

Thus similar matrices have the same spectrum,

$$\Lambda(A) = \Lambda(B), \tag{37}$$

with the same algebraic multiplicities. Next,

$$Ax = \lambda x \Leftrightarrow X^{-1}BXx = \lambda x \Leftrightarrow B(Xx) = \lambda(Xx). \tag{38}$$

This shows that the eigenspace E_λ of A with λ and the eigenspace E'_λ of B are related as follows: $E_\lambda = XE'_\lambda$. It follows that λ has the same geometric multiplicity relative to A and B.

Defective Matrices. Geometric multiplicity can be different from algebraic multiplicity. An eigenvalue λ is **defective** if its geometric multiplicity is different from its algebraic multiplicity. A matrix is **defective** if any of its eigenvalue is defective. For instance, if

$$A = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}, \qquad A' = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}, \qquad (\lambda \neq 0) \tag{39}$$

then $\Lambda(A) = \Lambda(A') = \{\lambda\}$ and the algebraic multiplicity of λ is 2 for both A and A'. Let E_λ and E'_λ be the eigenspaces associated with λ for A and A', respectively. Clearly, $E_\lambda = \mathbb{C}^2$ so that the geometric multiplicity of λ relative to A is 2. But if $x = (a, b)^T \in E'_\lambda$ then $A'x = \lambda x$. This means $\lambda x = (\lambda a + b, \lambda b)^T$, and thus $b = 0$. So E'_λ has dimension 1, not 2. Hence λ is defective for A'.

Lemma 6 *For all A, the geometric multiplicity of any $\lambda \in \Lambda(A)$ is at most the algebraic multiplicity of λ.*

Proof. To see this, suppose λ has geometric multiplicity m and $x_1, \ldots, x_m$ are m linearly independent eigenvectors all associated with λ. We may assume that the x_i's are unit vectors ($x_i^* x_i = 1$). Let $X = [x_1|x_2|\cdots|x_m|\cdots|x_n]$ where the columns $x_{m+1}, \ldots, x_n$ are additional unit vectors that span the complement of the eigenspace E_λ. Thus X is unitary ($X^*X = I$) and $X^{-1} = X^*$. So $AX = [\lambda x_1|\cdots|\lambda x_m|x'_{m+1}|\cdots|x'_n]$, where $x'_j = Ax_j$ for $j = m+1, \ldots, n$. Then

$$B = X^*AX = \begin{bmatrix} \lambda I & C \\ \mathbf{0} & D \end{bmatrix}$$

for some C and D. This shows that the characteristic polynomial of B is divisible by $(z - \lambda)^m$, and so the algebraic multiplicity of B is at least m. Since A and B are similar, the algebraic multiplicities λ in A and B are equal.
Q.E.D.

Diagonalizability. The diagonal matrix whose (i, i)th element is d_i (for $i = 1, \ldots, n$) is denoted $D = \text{diag}(d_1, \ldots, d_n)$. A matrix is **diagonalizable** if it is similar to a diagonal matrix. To see why this concept is useful, suppose A is diagonalizable:

$$A = XDX^{-1}, \quad D = \text{diag}(d_1, \ldots, d_n) \tag{40}$$

for some X. Then observe that the columns of X are eigenvectors for A. To see this, we have $AX = XD$ and hence $Ax_i = d_i x_i$ where x_i is the ith column of X. Furthermore, this set of eigenvectors is **complete**, i.e., they span the whole space.

We restate this observation as a theorem. Let e_i denote the ith **elementary vector**, with 1 in the ith position and 0 elsewhere. Thus $x_i = Xe_i$. and $d_i = De_i$.

Theorem 7 *If $A = XDX^{-1}$ where D is a $n \times n$ diagonal matrix, then the set $\{Xe_1, \ldots, Xe_n\}$ is a complete set of eigenvectors of A. Moreover, De_i is the associated eigenvalue of Xe_i.*

It follows that a *diagonal matrix $D = \text{diag}(d_1, \ldots, d_n)$ is always non-defective.* To see this, note that the characteristic polynomial of D is $\prod_{i=1}^n (x - d_i)$ and so each $\lambda \in \{d_1, \ldots, d_n\}$ is an eigenvalue of D and it appears in D as many times as its algebraic multiplicity. Since similarity transformations preserve eigenvalues and their multiplicities, we conclude:

Theorem 8 *A matrix A is diagonalizable iff it is non-defective.*

For instance, the matrix A' in (39) is not diagonalizable.

Unitary Similarity and Schur Form. Canonical form for matrices that are equivalent under various notions of equivalence is an extremely powerful tool in linear algebra. We will consider matrices that are equivalent under unitary similarity transformations: $A \equiv B$ iff $A = UBU^*$ for some unitary U. Unitary operators are basically isomorphisms of the inner product space: if $y = Ux$ then $y^*y = (Ux)^*(Ux) = x^*(U^*U)x = x^*x$.

The invariant subspace relationship can be captured by a matrix equation: let X be a $n \times m$ matrix whose m columns span some A-invariant subspace E. Then

$$AX = XB \tag{41}$$

where $B \in \mathbb{C}^{m\times m}$. Conversely, every such equation (41) shows that the space spanned by the columns of X is A-invariant.

Next, assume that $E \subseteq E_\lambda$ for some eigenvalue λ. Then B has the form λI in (41). Let $X = QR$ be a full QR-factorization of X (see (35)) where

$$R = \begin{bmatrix} T \\ \mathbf{0} \end{bmatrix}$$

for some upper triangular $T \in \mathbb{C}^{m\times m}$. To say that a matrix A is upper triangular means that $(A)_{ij} = 0$ for $j > i$. Thus (41) becomes $AQR = QRB$, or $Q^*AQR = RB$. Since $RB = \lambda R$, we have

$$(Q^*AQ)R = \lambda R. \tag{42}$$

Let us write

$$Q^*AQ = \begin{bmatrix} C & D \\ E & F \end{bmatrix}$$

where $C \in \mathbb{C}^{m\times m}$ and $F \in \mathbb{C}^{(n-m)\times(n-m)}$. Then the block version of (42) implies $ET = \mathbf{0}$. Since T is non-singular, this means $E = \mathbf{0}$:

$$Q^*AQ = \begin{bmatrix} C & D \\ \mathbf{0} & F \end{bmatrix}. \tag{43}$$

By repeated application of this transformation of A, we obtain the **Schur Decomposition** of a matrix:

Theorem 9 (Schur Decomposition) *Every matrix A is unitarily similar to a upper diagonal matrix T.*

Proof. We use induction on n where $A \in \mathbb{C}^{n\times n}$. The result is trivial for $n = 1$. So assume $n =\geq 2$ and let $Ax = \lambda x$ for some eigenvector $x \neq 0$. Then using (43) with $m = 1$, there is a unitary Q such that

$$Q^*AQ = \begin{bmatrix} c & y^T \\ \mathbf{0} & F \end{bmatrix}$$

for some $c \in \mathbb{C}$, some vector y and square matrix F. By induction, there is unitary V such that V^*FV is upper triangular. If

$$U = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & V \end{bmatrix}$$

then $T = U^*(Q^*AQ)U$ is upper triangular. As QU is unitary, this shows that A is unitarily similar to T. **Q.E.D.**

Unitary Diagonalizability. We have introduced two concepts: unitary similarity and diagonalizability. Combining them, we say a matrix A is **unitarily diagonalizable** iff it has the form

$$A = U\Lambda U^*$$

for some unitary U and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$.

Clearly, unitarily diagonalizability implies diagonalizability. We show the converse as well: let A be diagonalizable, so $A = X\Lambda X^{-1}$ for some X. Here X is not necessarily unitary. By theorem 7, the columns $\{Xe_1, \ldots, Xe_n\}$ of X forms a complete set of eigenvectors for A. In order for X to be unitary, we need the eigenvectors to be normalized, i.e., $(Xe_i)^*(Xe_i) = 1$. Let $w_i := \sqrt{(Xe_i)^*(Xe_i)}$ and $u_i :=(Xe_i)/w_i$. Next define the matrices U and W via

$$X = U \cdot W = [u_1|u_2|\cdots|u_n] \cdot \mathrm{diag}(w_1, \ldots, w_n).$$

Clearly, U is unitary and we have $X^{-1} = W^{-1}U^{-1} = W^{-1}U^*$. Of course, $W^{-1} = \mathrm{diag}(1/w_1, \ldots, 1/w_n)$. We then have $A = (UW)\Lambda W^{-1}U^* = U\Lambda U^*$. So A is unitarily diagonalizable. In other words: *a matrix is diagonalizable iff it is unitarily diagonalizable.*

Recall that a matrix A is normal if $A^*A = AA^*$. We note a useful lemma.

LEMMA 10 *If A is upper diagonal, then A is normal iff A is diagonal.*

Proof. One direction is easy: if A is diagonal, then clearly $A^*A = AA^*$. Conversely, suppose $A^*A = AA^*$. We claim that A must be diagonal. Let c be the first column of A and r be the first row of A. Then top-left corner entry of A^*A is c^*c, and the corresponding entry of AA^* is r^*r. Thus A is normal implies $c^*c = r^*r$. But the first entry in c and in r are equal to some $\alpha \in \mathbb{C}$, and $c^*c = |\alpha|^2$. Hence $r^*r = |\alpha|^2$, which implies that all the remaining entries in r are zero. Continuing in this fashion, we argue that all the off-diagonal entries in the ith row must be zero. **Q.E.D.**

THEOREM 11 *A matrix is normal iff it is diagonalizable.*

Proof. Let $A = UTU^*$ be the Schur decomposition of A given by the previous theorem. Since $AA^* = (UT)(T^*U^*)$ and $A^*A = (UT^*)(TU^*)$, we have

$$AA^* = A^*A \Leftrightarrow T^*T = TT^*.$$

Thus A is normal iff T is normal. But we had just shown that T is normal iff T is diagonal. Thus A is normal iff T is diagonal. But T is diagonal means A is diagonalizable. **Q.E.D.**

As a corollary, we have

- Unitary and Hermitian matrices are diagonalizable. This is because such matrices are normal. It follows that such matrices have complete sets of eigenvectors.
- A matrix is non-defective iff it is normal. This follows from theorems 11 and 8.

A **projection operator** P is characterized by the equation $P^2 = P$. For any unit length vector x, the matrix xx^* is a projection operator since $(xx^*)(xx^*) = x(x^*x)x^* = xx^*$. Note that xx^* is Hermitian, since $(xx^*)^* = xx^*$. For any y, $(xx^*)y = (x^*y)x = \alpha x$ where $\alpha = x^*y$ is a scalar. Thus the range of the projection (xx^*) is the linear subspace spanned by x. Generalizing this, if $\{x_i : i = 1, \ldots, k\}$ is a set of orthonormal vectors, then $P = \sum_{i=1}^{k} x_i x_i^T$ is a projection operator.

Hermitian Matrices and Real Numbers. We show a remarkable family analogies between Hermitian matrices and real numbers. A matrix A is **positive semidefinite** (resp., **positive definite**) if $x^*Ax \ge 0$ (resp., $x^*Ax > 0$) for all nonzero vector x. Analogous definitions can be given by replacing "positive" by "negative".

THEOREM 12 *Let H be Hermitian.*
(i) All its eigenvalues are real.
(ii) H is positive definite iff all its eigenvalues are positive.
(iii) H is positive semidefinite iff all its eigenvalues are non-negative.

Proof. Let $H = U\Lambda U^*$ for some unitary U and diagonal Λ.
(i) By the normality of H, we have $H = H^* = U\Lambda^*U^*$. Hence $\Lambda = \Lambda^*$, i.e., Λ is real.
(ii) Let the columns of U be $x_1, \ldots, x_n$. Each x_i is an eigenvector of H with associated eigenvalue λ_i. Since $x_i^*Hx_i = \lambda_i|x_i|^2$, the positive definiteness of H implies $\lambda_i > 0$. Conversely, if each $\lambda_i > 0$ we can show that H is positive definite: any non-zero vector $x \in \mathbb{C}$ can be expressed as $\sum_{i=1}^{n} c_i x_i$ ($c_i \in \mathbb{C}$). Then $x^*Hx = \sum_{i=1}^{n} \lambda_i |c_i|^2 |x_i|^2 > 0$
(iii) The proof is similar to (ii). **Q.E.D.**

Analogy between Hermitian Matrices and Real Numbers. The above connection between Hermitian matrices with real numbers goes much deeper. The special role of real numbers in the complex field $\mathbb{C}$ is mirrored in many ways by the Hermitian matrices in the context of complex matrices. This analogy can be extended as follows:

real number	$\leftrightarrow$	Hermitian
pure complex number	$\leftrightarrow$	anti-Hermitian
complex sign, $\lvert z\rvert = 1$	$\leftrightarrow$	unitary
positive real	$\leftrightarrow$	positive definite Hermitian
non-negative real	$\leftrightarrow$	positive semidefinite Hermitian

In the following, z is a complex number. The complex conjugate of $z = x + \mathbf{i}y$ is $\overline{z} = x - \mathbf{i}y$. We point out the matrix analogues of the following properties:

1. z is real iff $z = \overline{z}$; z is pure complex iff $\overline{z} = -z$.
2. z is real iff $\mathbf{i}z$ is pure complex; z is pure complex iff $\mathbf{i}z$ is real.
3. $z + \overline{z}$ is real and $z - \overline{z}$ is pure complex.
4. $\overline{z}z = z\overline{z}$ is real.
5. z can be uniquely written as $z = x + \mathbf{i}y$ where x, y are real.
6. z can be uniquely written as $z = x + w$ where w is real and w is pure complex.
7. A real number r is non-negative iff there is a real number s such that $r = s^2$.
8. A z has the **polar form**, $z = rs$ where r is non-negative real, and s is a complex sign, $|s| = 1$. This form is unique if z is non-zero.
9. A complex sign s can be uniquely written as $s = e^{\mathbf{i}\theta}$ for some real θ.

In the following, let $A \in \mathbb{C}^{n\times n}$.

1. A is Hermitian iff $A = A^*$; A is anti-Hermitian iff $A^* = -A$.
   Thus, the matrix analogue of complex conjugation, $z \mapsto \overline{z}$, is conjugate transpose, $A \mapsto A^*$.
2. A is Hermitian iff $\mathbf{i}A$ is anti-Hermitian; A is anti-Hermitian iff $\mathbf{i}A$ is Hermitian.
3. $A + A^*$ is Hermitian and $A - A^*$ is anti-Hermitian.
4. A^*A and AA^* are both Hermitian.
5. A can be uniquely written as $G + \mathbf{i}H$ where G, H are Hermitian.
6. A can be uniquely written as $A = G + F$ where G is Hermitian and F is anti-Hermitian.
   This is just a restatement of the previous property.
7. A Hermition matrix H is positive semidefinite iff $H = G^*G$ for some positive semidefinite G.
8. A has two polar forms, $A = HU$ and $A = U'H'$, where H, H' are positive semidefinite Hermitian, and U, U' are unitary. If A is non-singular, then these polar forms are unique.
9. A unitary A can be uniquely written as $e^{\mathbf{i}H}$ for some Hermitian H.

Let us prove the non-obvious cases of the properties.
Property 5: Let $G = (A + A^*)/2$ and $H = (A - A^*)/2\mathbf{i}$. Then clearly $A = G + \mathbf{i}H$ and from the preceding, we conclude that G and H are Hermitian. Conversely, if $A = G + \mathbf{i}H$ for some Hermitian G and H, then $A^* = G^* + (\mathbf{i}H)^* = G^* - \mathbf{i}H^* = G - \mathbf{i}H$. It follows that $G = (A + A^*)/2$ and $H = (A - A^*)/2\mathbf{i}$.
Property 7: If A is positive semidefinite, we know that $A = U\Lambda U^*$ for some unitary U and diagonal $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_i > 0$. If $B = U\mathrm{diag}(\sqrt{\lambda_1}, \ldots, \mathrm{diag}\sqrt{\lambda_n}U^*$, then $BB^* = A$.
Property 8:
Property 9:

Jordan Form. The ultimate canonical form under general similarity transformation is the Jordan form.

Hilbert Space. Let S be a complex vector space (or linear space), endowed with an inner product $\langle x, y\rangle \in \mathbb{C}$ such that for all $x, y, z \in S$ and $a \in \mathbb{C}$,

- (linearity) $\langle x, y + z\rangle = \langle x, y\rangle + \langle x, z\rangle$
- (homogeneity) $\langle x, ay\rangle = a\langle x, y\rangle$, $a \in CC$.
- (skew symmetry) $\langle x, y\rangle = \overline{\langle y, x\rangle}$
- (positivity) $\langle x, x\rangle$ is real and $\langle x, x\rangle \geq 0$ with equality iff $x = 0$

There is an asymmetry in the two arguments of the inner product: it follows from the above axioms that

$$\langle ax, y\rangle = \overline{\langle y, ax\rangle} = \overline{a}\overline{\langle y, x\rangle} = \overline{a}\langle x, y\rangle.$$

How does such inner products arise? Suppose H is a matrix and we define $\langle x, y\rangle$ to be x^*Hy. Linearity and homogeneity are obvious: $x^*H(y+z) = (x^*Hy) + (x^*Hz)$ and $x^*H(ay) = a(x^*Hy)$. If H is Hermitian then skew symmetry holds: $\overline{\langle y, x\rangle} = (y^*Hx)^* = x^*H^*y = x^*Hy = \langle x, y\rangle$. It follows from theorem 12 that if H is positive definite then positivity also holds. The simplest case is to choose H to be the identity matrix I.

The **norm** $\|x\|$ of $x \in S$ is defined to be $\sqrt{\langle x, x\rangle}$. An infinite sequence $\overline{x} = (x_1, x_2, \ldots, x_k, \ldots)$ is **Cauchy** if for every $\varepsilon > 0$ there is some k such that for all $i, j \geq k$, $\|x_i - x_j\| < \varepsilon$. The limit of $\overline{x}$ is the element $x_0 \in S$ such that $\|x_k - x_0\| \to 0$ as $k \to \infty$ (clearly, x_0 is unique). The space S is **complete** if every Cauchy sequence has a limit. A **Hilbert space** is a complex vector space S with an inner product and norm as defined, and which is complete. The literature sometimes require Hilbert space to be infinite dimensional; for our limited purposes, we will actually assume S is finite dimensional. The infinite dimensional setting is somewhat more complicated. For instance, in the finite dimensional setting, if an operator A satisfies $A^*A = I$ (i.e., it is unitary), then $AA^* = A^*A$. This property may fail in the infinite dimensional setting.

Duality. For each $x \in S$ we obtain a linear function $f_x : S \to \mathbb{C}$ where $f_x(y) = \langle x|y\rangle$:

$$f_x(cy) = cf_x(y), \qquad f_x(y+z) = f_x(y) + f_x(z).$$

The function f_x is also continuous where the underlying topology on S is given by the metric $d(x, y) = \|x - y\|$. Conversely, if $\phi : S \to \mathbb{C}$ is linear and continuous, there exists $y \in S$ such that $\phi = f_y$. This duality between elements of S and continuous linear functions on S gives rise to the $|x\rangle$ (this is just x) and $\langle y|$ notation (this is f_y). Moreover, $\langle y||x\rangle = f_y(x) = \langle y, x\rangle$.

Exercise

Exercise 22.5.1: Show that $\det(AB) = \det(A)\det(B)$. □

Exercise 22.5.2: Show if $AB = I$ then $BA = I$ and B is unique. HINT: let the (i, j)**-cofactor** of A be $(-1)^{i+j}$ times the determinant of the matrix A with the ith row and jth column deleted. Consider the matrix C whose (i, j)th entry is the (j, i)-cofactor. How close is C to the inverse of A? Show that $AC = CA$. □

Exercise 22.5.3: Show a positive definite matrix that is not Hermitian. □

Exercise 22.5.4: Let C_{ij} denote the (i, j)-cofactor of A, defined to be $(-1)^{i+j}$ times the determinant of A after the i-th row and j-th column is deleted. Assume the following fact: for all i, $\det(A) = \sum_{j=1}^{n} a_{ij}C_{ij}$ where $a_{ij} = (A)_{ij}$.
(i) Prove that if $\det(A) \neq 0$ then there exists B such that $AB = BA = I$. HINT: Consider the "adjoint" $adj(A)$ of a matrix A where the (i, j)-th entry of $adj(A)$ is the (j, i)-cofactor C_{ji} (note the transposed subscripts).
(ii) Assume $AB = BA = I$. Suppose $AB' = I$ or $B'A = I$ for some other B'. Prove that $B = B'$. [From (i) and (ii), we conclude that inverses are defined and unique whenever $\det A \neq 0$. □

Exercise 22.5.5: Consider $n \times n$ matrices with complex entries which are either orthogonal or unitary.
(i) If $n = 1$, what do these matrices look like?
(ii) If $n = 2$, what do these matrices look like? □

Exercise 22.5.6: Let $\lambda_1, \ldots, \lambda_k$ be distinct eigenvalues of A and for $i = 1, \ldots, k$, B_i is a set of linearly independent vectors of the invariant subspace E_{λ_i}. Then the set $B = \cup_{i=1}^{k} B_i$ is linearly independent. □

Exercise 22.5.7: (SVD) The singular value decomposition (SVD) of an $m \times n$ matrix A is $A = U\Sigma V$ where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are both unitary, and Σ is diagonal. The diagonal entries of Σ are called the **singular values** of A. Show that every A has a SVD. Further, up to complex signs and ordering of the singular values in Σ, the columns of U and rows of V are unique. □

Exercise 22.5.8: A polynomial $q(z)$ is a **minimal polynomial** for a matrix A if $q(A) = 0$ and q has minimal degree.
(i) Show that $q(z)$ divides the characteristic polynomial of A.
(ii) Characterize the matrices $A \in \mathbb{C}^{n\times n}$ such that the minimal polynomial of A is z^n □

End Exercise

Bibliography

[1] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52(5):3457–3467, 1995.

[2] D. Beckman, A. N. Chari, S. Devabhakturi, and J. Preskill. Efficient networks for quantum factoring. *Phy. Rev. A*, 54(2):1034–1063, 1996.

[3] P. Benioff. The computer as a physical system: A macroscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *J. Stat. Phys.*, 22(5):563–590, 1980.

[4] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines. *J. Statist. Phys.*, 29:515–546, 1982.

[5] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines that dissipate no energy. *Phys. Review Letters*, 48:1581–1585, 1982.

[6] P. Benioff. Quantum ballistic evolution in quantum mechanics: Applications to quantum computers. *Phy. Rev. A*, 54(2):1106–1123, 1996.

[7] C. H. Bennett. Logical reversibility of computation. *IBM J. Res. Develop.*, 17:525ff, 1973.

[8] C. H. Bennett. Notes on the history of reversible computation. *IBM J. Research and Develop.*, 32:16–23, 1988.

[9] C. H. Bennett. Time/space trade-offs for reversible computation. *SIAM J. Computing*, 18:766–776, 1989.

[10] C. H. Bennett and P. W. Shor. Quantum information theory. *IEEE Trans. Info. Theory*, 44:2724–2742, 1998.

[11] E. Bernstein and U. Vazirani. Quantum complexity theory. *Proc. ACM Symposium on Theory of Computing*, 25:11–20, 1993.

[12] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Physical Review Letters*, 74(20):4091ff, 1995.

[13] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proc. Royal Soc. London, A*, 454:339–354, 1998.

[14] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Royal Soc. London, A*, 400(97–117), 1985.

[15] D. Deutsch. Quantum computational networks. *Proc. Royal Soc. London, A*, 439:553–558, 1992.

[16] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. Royal Soc. London, A*, 439(553–558), 1992.

[17] R. P. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507–531, 1986. Reprinted from Opt.New Vol.11, 11(1985).

[18] E. Fredkin and T. Toffoli. Conservative logic. *Int. J. Theor. Phys.*, 21:219–253, 1982.

[19] Y. Lecerf. Machines de Turing réversibles. récursive insolubilté en $n \in \mathbb{N}$ de l'équation $u = \theta^n u$, oú θ est un isomorphisme de codes. *C. R. Acad. Fran caise Sci.*, 257:2597–2600, 1963.

[20] C. Monroe, D. Meekhof, B. King, W. Itano, and D.J.Wineland. Demonstration of a universal quantum logic gate. *Physical Review Letters*, 75:4714ff, 1995.

[21] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[22] M. Raizen, J. Gilligan, J. Bengquist, W. Itano, and D. Wineland. Ionic crystals in a linear paul trap. *Phy. Rev. A*, 45:6493ff, 1992.

[23] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Computing*, 26(5):1484–1509, 1997.

[24] T. Toffoli. Reversible computing. In J. de Bakker and J. van Leeuwen, editors, *Proc. 7th Int. Colloquium on Automata, Languages and Programming*, pages 632–644, New York, 1980. Springer. Lecture Notes in Computer Science, vol.84.